UNIVERSIDAD RICARDO PALMA

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

PROTOTIPO PARA MONITOREAR LA POSICIÓN DE BEBÉS EN SUS CUNAS, UTILIZANDO VISIÓN ARTIFICIAL

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO

PRESENTADA POR:

Bach. VIDAL LORA, CRISTOPHER DARÍO

ASESOR: Dr. Ing. HUAMANÍ NAVARRETE, PEDRO FREDDY

LIMA – PERÚ

2020

DEDICATORIA

Dedico la presente investigación a todas esas personas que me han ofrecido su apoyo incondicional y desinteresado, por cada minuto entregado y sobre todo por el voto de confianza depositado en mí. A mi madre y a mi padre por estar siempre pendientes de mi futuro. No hay forma adecuada para agradecer todo lo que han hecho por mí.

AGRADECIMIENTO

Siempre tener en cuenta a los docentes de la carrera que brindaron herramientas, experiencia y el conocimiento necesario para forjar nuevos profesionales; a la casa de estudios, la Universidad Ricardo Palma, y compañeros de estudio con quienes aprender se hizo más ameno.

Familiares y amigos que siempre están dispuestos a ofrecer su apoyo y motivación.

Y, el más grato agradecimiento a todos ellos por permitirme hoy llevar el título profesional de Ingeniero Electrónico.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ANEXOS

RESUMEN

El presente trabajo de tesis explica la integración de los controladores Raspberry y Arduino, los cuales cuentan con un procesador ARM Cortex-A53 y un controlador ATMEGA 328P, respectivamente. De esta manera, cada tarjeta electrónica realizó procesos independientes, siendo el Raspberry utilizado para el procesamiento y el ATMEGA 328P para controlar los actuadores y adquisición de peso de la cuna. Además, estos dispositivos se comunican de manera serial mediante la configuración de Maestro-Esclavo. Asimismo, el prototipo desarrollado en este proyecto de tesis tuvo en cuenta el soporte para la cámara de video encargada de monitorear el área de la cuna del bebé, permitiendo de esta manera registrar cada cierto tiempo su posición y realizar una acción correspondiente a cada caso de detección del mismo. Asimismo, esta detección fue realizada mediante una cámara conectada al Raspberry, la cual una vez que identificó la posición, mantuvo en alerta al Arduino conectado a un periférico distractor; asimismo, al módulo GSM SIM900. Por lo cual, en el caso se detecte al bebé en una posición de riesgo, el Raspberry envía de manera automática al Arduino la instrucción para realizar el envío de un mensaje de texto al supervisor del bebé, con el fin de alertarlo sobre la condición actual del mismo. De esta manera se logra mantener una supervisión directa del bebé en su cuna, brindando a los padres y/o apoderados la tranquilidad de seguridad con la implementación de este prototipo. Finalmente, en las pruebas el prototipo demostró ser confiable y se obtuvo un margen de error del 35% con posiciones no registradas.

Palabras claves: Raspberry, Arduino, procesamiento de imagen, microcontroladores, microprocesadores

ABSTRACT

This thesis work explains the integration of Raspberry and Arduino controllers, which have an ARM Cortex-A53 processor and an ATMEGA 328P controller, respectively. In this way, each electronic card carried out independent processes, the Raspberry being used for processing and the ATMEGA 328P to control the actuators and the acquisition of cradle weight. In addition, these devices communicate serially through the Master-Slave configuration. In the same way, the prototype developed in this thesis project took into account the support for the video camera in charge of monitoring the area of the baby's cradle, they controlled this position every so often their position and performs an action corresponding to each case of detection thereof. Similarly, this detection was carried out by means of a camera connected to Raspberry, which once it identified the position, alerts alerted the Arduino connected to a peripheral distractor; specifically, to the GSM SIM900 module. Therefore, in the case the baby is detected in a risky position, the Raspberry automatically sends the instruction to send a text message to the baby's supervisor to the Arduino, in order to alert the current condition of the same In this way it is possible to maintain a direct supervision of the baby in his cradle, providing the parents and / or guardians the peace of mind with the implementation of this prototype. Finally, in the tests the prototype proved to be reliable and a margin of error of 35% was obtained with unregistered positions.

Keywords: Raspberry, Arduino, Image processing, microcontrollers, microprocessors.

INTRODUCCIÓN

El presente trabajo de investigación se presenta al lector como un prototipo electrónico para la supervisión de un bebé en su cuna mientras se encuentra dormido y en la ausencia de una persona responsable de su cuidado. En la actualidad, padre y madre suelen trabajar, por lo que ambos se ausentan en casa, por ello muchas veces se contrata a una persona para brindar cuidado y atención a su hijo, y con especial énfasis cuando se trata de un bebé; por ello, lo que los padres buscan es tener la seguridad de poder salir a trabajar sabiendo que su hijo se encuentra en buenas manos; así mismo, el poder realizar otras actividades mientras dejan a su bebé dormido en su cuna. Sin embargo, mientras el bebé descansa existe un riesgo potencial, ya que estudios médicos revelan que la mala postura de un bebé al dormir puede conllevar a la asfixia o atragantamiento. Además, es necesario tener en cuenta que cuando los bebés tienen la fuerza suficiente para poder pararse, son propensos a sufrir golpes por caídas; además, considerando la curiosidad que poseen, estos suelen "morder" las barandas hasta el punto en que llegan a ingerir partículas de pintura y hasta incluso madera.

Por lo tanto, el desarrollo de la presente investigación trata de un prototipo para monitorear la posición de bebés en sus cunas utilizando visión artificial, y para ello se tiene como antecedente a los siguientes autores: Wei-Liang Ou en su investigación sobre la detección de objetos extraños, vómito y gestos faciales basados en video, en la cual propone el uso de tecnología "inteligente" diseñada en forma de reloj para el monitoreo. Fatih Elmas, en su investigación sobre el modelado y prototipo del diseño automático de la horquilla de oscilación de una cuna, basada en la repetición y simulación de los movimientos de una madre. Finalmente, Soukaina Brangui con su sistema integral ergonómico para el monitoreo y control de ambientes para bebés, el cual ubica sensores de manera estratégica para llevar a cabo el control del ambiente del bebé. Tal como se puede apreciar, las investigaciones mencionadas se enfocan en el cuidado de un bebé en interiores y utilizan técnicas especializadas para conseguir los propósitos deseados.

De esta manera, el problema general de esta investigación radica en cómo implementar un prototipo electrónico con apoyo de técnicas de visión artificial para monitorear la posición de un bebé en su cuna. Y, como motivo principal, se tiene a la supervisión del bebé mientras duerme, así como generar alertas cuando este se encuentre en una posición de riesgo, para que los padres y/o responsables de su cuidado acudan y eviten accidentes.

Así mismo, los resultados obtenidos de este trabajo servirán como base para la mejora del diseño algorítmico e implementación de sistemas más complejos en cuanto al cuidado, alerta y sensado enfocado a bebés.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1. Fundamentación y formulación del problema

En la actualidad, el índice de accidentes de bebés se genera en la mayoría de las veces por descuidos de los padres y/o apoderados, al no tener a sus pequeños bajo un monitoreo constante mientras se encuentran en la cuna; principalmente cuando se deja al bebé durmiendo en su habitación y este despierta por un motivo desconocido. Por ello, esto puede generar que el bebé se ponga de pie al filo de la baranda, caiga y se lastime. Así como también, al encontrarse dormido, el bebé puede girar y colocarse en una posición de riesgo para su salud.

Por lo tanto, este proyecto de tesis otorga una mayor confiabilidad al dejar a los bebés solos mientras duermen en la cuna de su habitación, brindando de esta manera tranquilidad y confianza a los padres. Pues, este prototipo electrónico está conformado por una tarjeta integrada Raspberry Pi modelo 3B, programada con OpenCV para implementar técnicas de procesamiento de imágenes. De esta manera, su uso resulta sencillo para el usuario, ya que es suficiente la instalación de los equipos correspondientes para que los padres reciban vía mensaje de texto en sus teléfonos móviles, los eventos potencialmente peligrosos que ocurren en el cuarto del infante. Entonces, para este propósito se tiene implementado un sistema de telecomunicaciones vía GSM, que ofrece una rápida alerta si es que el bebé se encuentra en peligro potencial, para que los padres o la persona a cargo puedan actuar a tiempo y acudir al cuidado del menor. Por otro lado, en el caso del uso de los sensores y la cámara de video, se justifica su utilización debido a que mantiene un constante reconocimiento del estado del bebé, al evaluarse el peso al borde de la cuna cuando el menor se encuentra muy cerca de la baranda; asimismo, cuando el infante se despierte será posible activar un juguete móvil distractor y evitar que llore. La justificación principal del proyecto recae en que existe una alta tasa de mortalidad en infantes mientras duermen en sus cunas debido a la falta de una adecuada supervisión, por eso se busca mantener en buen estado al infante y aliviar a los padres manteniéndolos al tanto sobre el estado de su bebé.

1.1.1 Problema principal

¿Cómo desarrollar un prototipo electrónico para monitorear la posición de bebés en sus cunas, a partir de la implementación de técnicas de visión artificial?

1.1.2 Problemas secundarios

a) ¿Cómo implementar un prototipo electrónico de monitoreo diurno en tiempo real, utilizando un Raspberry Pi, un Módulo GSM y una cámara de video?

b) ¿Cómo implementar algorítmicamente técnicas de visión artificial en un Raspberry Pi, utilizando las librerías de Open CV, para reconocer la posición del bebé en una cuna de tamaño estándar?

c) ¿Cómo generar alertas de mensajes de texto ante un posible peligro del bebé mientras se encuentra en su cuna, así como también tomar acción con juguetes distractores?

1.2. **Objetivo general y específicos**

1.2.1 Objetivo general

Desarrollar un prototipo electrónico para monitorear bebés en sus cunas, a partir de la implementación de técnicas de visión artificial.

1.2.2 Objetivos específicos

a) Implementar un prototipo electrónico de monitoreo diurno en tiempo real, utilizando un Raspberry Pi, un Módulo GSM y una cámara de video.

b) Implementar algorítmicamente técnicas de visión artificial en un Raspberry Pi, utilizando las librerías de Open CV, para reconocer la posición del bebé en una cuna de tamaño estándar.

c) Generar alertas de mensajes de texto ante un posible peligro del bebé mientras se encuentra en una cuna de tamaño estándar, así como también tomar acción con juguetes distractores.

1.3. **Delimitación de la investigación: Temporal, Espacial y Temática**

a) Temporal: El proyecto se desarrolló en un tiempo limitado de aproximadamente 9 meses.

b) Espacial: El proyecto de tesis es aplicable en bebés dentro del rango de edad de 6 meses a 2 años.

c) Temática: Existen sistemas de monitoreo que son exportados al Perú para su venta; pero ninguno utilizando Raspberry Pi como controlador principal.

1.4. **Justificación e importancia**

Este proyecto de tesis es importante porque brinda una mayor seguridad al dejar a los bebés solos mientras duermen en la cuna de su habitación, ofreciendo tranquilidad y confianza a los padres, debido a que el prototipo ofrece confiabilidad y estabilidad en

el proceso de monitoreo. Asimismo, desde el punto de vista técnico, se justifica porque para su implementación se necesitó del Raspberry Pi para procesar las imágenes digitales con apoyo de la librería OpenCV; además, los padres no necesitan aprender comandos complejos para la utilización del prototipo, debido a que la integración de los equipos correspondientes permite el envío de mensajes de texto a sus teléfonos móviles vía la tecnología GSM, informando sobre los eventos de riesgo a los cuales está sometido el bebé en su cuarto. Y como también, el uso de los sensores de peso y la cámara de video, se justifican porque mantienen un constante reconocimiento del estado del niño cuando se pone de pie o se aproxime demasiado a las barandas de la cuna, emitiendo una alerta.

Por otro lado, la necesidad de la realización de este proyecto se debe a que existe una alta tasa de peligro en bebés mientras duermen en sus cunas, pues no poseen una adecuada supervisión; además, cuando los bebés adquieren fuerza en los brazos, logran ponerse en pie por su propia cuenta aproximándose a las barandas de la cuna, y como no cuentan con la fuerza y el equilibrio suficiente con las piernas es probable que se caigan y se provoquen golpes en distintas partes de su cuerpo.

CAPÍTULO II: MARCO TEÓRICO

2.1 Antecedente del estudio de investigación

Como era de esperarse, ya se han generado investigaciones previas para el monitoreo del estado de un bebé en su cuna o habitación. A continuación, se ofrece una breve descripción de cada investigación. El caso 1 corresponde a Ou, quien ofrece la detección de objetos extraños, de vómitos y gestos faciales basados en vídeo para la vigilancia y seguridad del bebé, el método propuesto propuso desarrollar una tecnología inteligente diseñada en forma de reloj para el cuidado del bebé (Ou, 2013). El caso 2 es el investigado por Elmas, un diseño que permite el modelado y prototipo del diseño automático de la horquilla de oscilación de una cuna, mediante un diseño que repite y simula los movimientos de la madre (Elmas, 2017). El caso 3 planteado por Brangui, que es un sistema integral ergonómico para el monitoreo y control de ambientes para bebés, el cual utiliza diversos sensores de humedad y temperatura (Brangui, 2015). Comparando las investigaciones, en una primera instancia se aprecian sus similitudes, es decir que sirven para el cuidado de bebés en interiores y utilizan técnicas especializadas para conseguir los propósitos deseados. De acuerdo con las investigaciones correspondientes a los casos 1 y 3, ambos ofrecen el monitoreo constante de la habitación del infante mientras este se encuentre solo. Por otro lado, para los casos 2 y 3 encontramos técnicas similares con el uso de sensores para la adquisición de datos (Ou, 2013), (Elmas, 2017); mientras que el caso 1 utiliza un algoritmo de detección en la región bucal y características del ojo para detectar condiciones anormales (Brangui, 2015). Sin embargo, las diferencias de implementación son notables, pues el caso 1 necesita que el reloj del bebé tenga filtros de ojo y de nodo cruzado, los cuales son propuestos para detectar la zona ocular y las características oculares. En segundo lugar, según las características del ojo y sus posiciones, se encuentra la región correcta de la boca por el método geométrico y se detecta condiciones peligrosas alrededor de la boca. (Ou, 2013) Luego, para el caso 2, se necesita en primer lugar que el ruido en los datos sea filtrado con el filtro media móvil para luego realizar la integración trapezoidal, la cual es aplicada dos veces a dichos datos filtrados de la aceleración. Después de eso, la ruta ideal para el movimiento de balanceo en la vuelta fue alcanzada en la consideración de las trayectorias de los temas. De esta manera, el modelo matemático de la ruta ideal fue derivado utilizando el método de la guarnición de la curva cuadrada, donde

la horquilla que sigue el modelo matemático fue diseñada en el ambiente de SolidWorks. Además, en la fase de modelado de la cuna, se experimentaron 10 sujetos (Elmas, 2017). Finalmente, dado el caso 3, se sugiere la integración de un subsistema de cancelación de ruido para un esquema ergonómico que está monitoreando y controlando las condiciones de la sala de un bebé, debido a que los sistemas existentes sólo monitorean y controlan la temperatura, humedad y luz en las habitaciones para bebés (Brangui, 2015). Ahora bien, las fortalezas de cada investigación se orientan a proporcionar una nueva tecnología en relojes de monitoreo, reflejar el movimiento de una madre mediante una aplicación IOS o Android en una cuna y agrupar diversos sensores para monitorear el estado de un bebé en su habitación; esto para el caso 1, caso 2 y caso 3, respectivamente. La debilidad del caso 1 es cuando el niño se encuentra boca abajo, pues no se puede realizar monitoreo alguno (Ou, 2013). La debilidad del caso 2 es que necesariamente alguien tiene que estar controlando cómo se mueve la cuna mediante un dispositivo móvil (Elmas, 2017). Finalmente, la debilidad del caso 3 es que solo pretende el sensado de parámetros alrededor del infante mas no se interesa en el estado mismo del bebé (Brangui, 2015).

Como se puede apreciar, no es la primera ni última vez que se han de realizar investigaciones sobre cómo monitorear en tiempo real y brindarle confort a un bebé mientras duerme o se encuentra en su cuna. En la presente investigación, se utilizaron también sensores; sin embargo, estos son controlados por el circuito integrado Raspberry Pi, ofreciendo confiabilidad y estabilidad. La originalidad de esta investigación se basa en fusionar sensores con una programación conjunta a la detección de posición del infante en la cuna, mediante imágenes a partir de una cámara y librerías de OpenCV que permite conocer si el bebé se encuentra parado en la baranda, o en qué posición se encuentra dormido, para alertar a los padres con una comunicación GSM la cual posee una cobertura amplia y no tarda más de 4 segundos en ser recepcionada.

2.2 Bases Teóricas vinculadas a las variables de estudio

2.2.1 Procesamiento de Imágenes

Se entiende por visión artificial a la adquisición, procesamiento, clasificación y reconocimiento de imágenes digitales, a partir de una cámara digital. Además, el píxel es el elemento básico de una imagen; y, a la imagen, se le considera como un arreglo bidimensional de píxeles con

diferente intensidad luminosa (escala de gris). En la Figura 1 se aprecia una imagen de 16 píxeles.

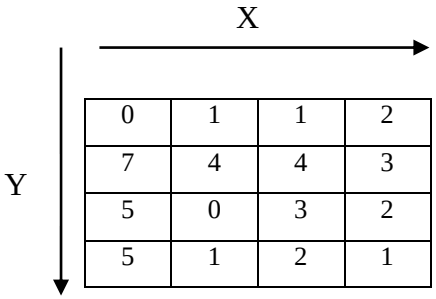


Figura 1: Imagen de 16 píxeles

Fuente: Elaboración propia

Además, si la intensidad luminosa de cada píxel se representa por "n" bits, entonces existirán 2^n escalas de gris diferentes. Entonces, matemáticamente una imagen digital se representa por r = f(x,y), donde r es la intensidad luminosa del píxel cuyas coordenadas son (x,y). Asimismo, un sistema de procesamiento de imágenes se representa como g(x,y) = T(f(x,y)) (Esqueda, 2002).

Además, la visión artificial tiene como fin el de simular los procesos visuales del hombre e intentar llegar a analizarlos por medio de un "cerebro", semejante al de los seres humanos. Pero, si se tiene la premisa que los hombres tienen la capacidad de transmitir imágenes captadas a través de la vista y analizarlas utilizando pulsos que se dirigen al cerebro, una máquina podría utilizar una cámara y estar habilitada para captar imágenes y enviarlas a un procesador, el cual estaría listo para analizarlas. De esta forma, la máquina podría inspeccionar el color y/o la forma de ciertos objetos (García, 2011). Es bueno resaltar que parece una hazaña simple el hecho de identificar un objeto o patrón específico desde una cámara y un procesador; sin embargo, en la práctica resulta exigente la programación que se ha de necesitar (Gimeno, 2010). Ahora bien, es necesario comentar un poco acerca del procesamiento de imágenes, ya que todos los sistemas de visión artificial necesitan del procesamiento y tratado de imágenes para poder lograr el objetivo trazado. Así como también, es habitual que las etapas aplicadas en dicho procesamiento digital de imágenes sean invariables para diversos softwares (Escolano, 2003). En el

caso de la adquisición de imágenes o captura, se requiere de un sensor y un digitalizador; ya que el proceso trata de convertir a un objeto en una representación capaz de ser procesada por una computadora. Esta adquisición la realiza un escáner, cámara fotográfica o de video. Por ello, es necesario seleccionar el tipo de cámara, distancia al objeto, método de codificación, etc. (Valle, 2015). Asimismo, el pre-procesamiento de la imagen comienza luego de haber obtenido la imagen digital, las operaciones y técnicas que se realizan y la obtención del resultado final en una nueva imagen. Por ello, el valor que tendrá el píxel de la imagen resultante es variante, ya que puede encontrarse en función de la imagen inicial, del valor de los pixeles vecinos, o de todos los pixeles en conjunto. Además, la segmentación consta en dividir la imagen en regiones manipulables, aquí se centra la base teórica, en otras palabras, se reconoce la información que se desea obtener, disgregando propiedades con una textura en particular. Se trabaja con la imagen en escala de grises y se utiliza operadores diferenciales para detectar los cambios de gradiente en dichos niveles de gris. La extracción de características usa el algoritmo que permitirá darle forma definitiva a la selección previa; en otras palabras, se extraen las características necesarias para la identificación de lo que se desea (Rojas, 2008). Finalmente, el reconocimiento e interpretación es comparado con una base de datos o algoritmo que está diseñado para tomar decisiones.

De esta manera, para mejorar la calidad o facilitar la búsqueda de información, existen diversas técnicas para el procesamiento de imágenes, tales como:

Proceso de Filtrado: Mejora las características de una imagen para una aplicación específica, generalmente se busca suavizar la imagen para disminuir la cantidad de variaciones en la intensidad de píxeles vecinos, eliminar el ruido quitando píxeles que poseen un nivel de intensidad muy diferente al de sus vecinos, realzar bordes, o detectar bordes identificando los cambios bruscos de la intensidad.

Retoque fotográfico: Proporciona una imagen modificada utilizando técnicas que mejoran la calidad de la imagen original luego de ser procesada (Gimeno, 2010).

Entonces, resumiendo la información anterior, se aprecia en la Figura 2 las etapas del tratamiento de imágenes y su orden respectivo. Esta información es necesaria para poder generar un diseño adecuado del algoritmo de procesamiento de imágenes, y así aumentar la probabilidad de éxito en reconocer la posición del bebé con el presente prototipo.

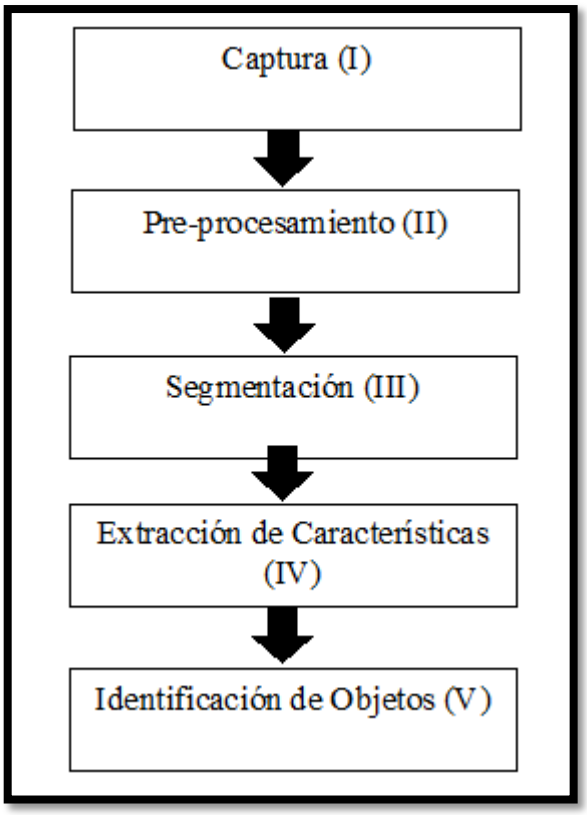


Figura 2: Etapas del tratamiento de imágenes

Fuente: Elaboración propia

La Figura 3 muestra como ejemplo el resultado del procesamiento de imagen de un bebé en su cuna, pero utilizando el método de la Transformada de Hough.

Sin embargo, el presente proyecto de investigación utiliza el método de procesamiento digital llamado Template Matching, para buscar y encontrar la localización/posición de una imagen dentro de otra. Es decir, encuentra partes menores de una imagen. Un ejemplo se denota en la Figura 4.

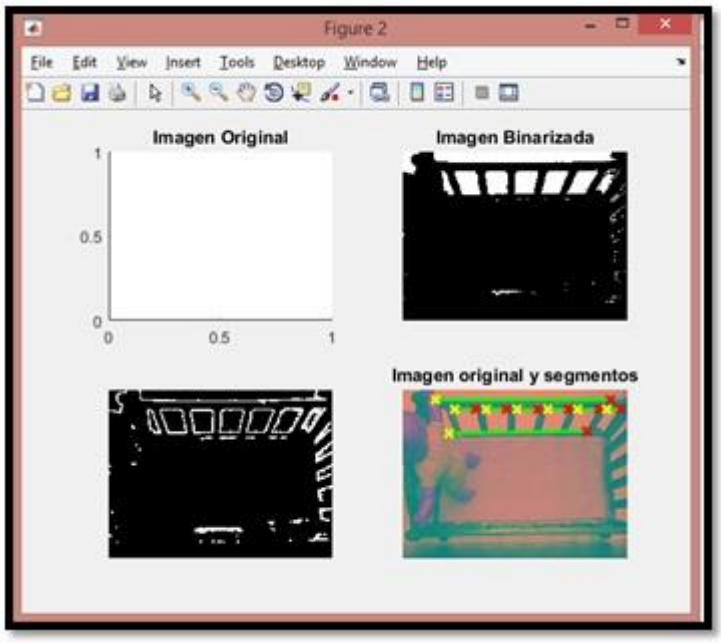


Figura 3: Ejemplo de procesamiento de una imagen

Fuente: Elaboración propia

Figura 4: Ejemplo del uso de la función cv2.minMaxLoc()

Fuente: Crespo, R. (2012). Opencv: Tratamiento de imágenes. Página 52

En lo que respecta a la librería OpenCV, este cuenta con una función llamada cv2.matchTemplate, el cual se basa en un algoritmo que posiciona una imagen denominada "template" sobre la imagen original, algo semejante a una convolución 2D, pero comparando la plantilla "template" con cada zona de la imagen original. Como resultado de ello se tiene una imagen a escala de grises, donde cada píxel indica cuánto es la aproximación de las imágenes. Luego, con la ayuda de la función

"cv2.minMaxLoc()" se busca donde se encuentra el máximo de parentesco entre las imágenes con el fin de localizar la zona en la que se desea centrar o analizar (Ou, 2013). En la Figura 4 se aprecia el resultado de emparejamiento de pixeles aproximados con el uso de la técnica Template Matching.

2.2.2 Comunicación Serial

Permite la transmisión de un flujo de bits serial de cualquier longitud sin implicar límites de caracteres. Implica al menos de dos estaciones participantes. La estación que tiene la responsabilidad para el enlace de datos y que emite comandos para controlar el enlace, se llama estación primaria. La otra es una estación secundaria. La comunicación de este enlace se establece de la estación primaria a la secundaria o viceversa (Morris, 2008).

2.2.3 Tecnología GSM

Es un sistema de telecomunicaciones digitales celulares normalizado por el Instituto Europeo para la Normalización en Telecomunicaciones (ETSI). Esta red proporciona enlaces de comunicación entre usuarios del servicio de telecomunicaciones móviles, incluso si se encuentran en células distintas o incluso en el dominio de diferentes operadores, así como el servicio de comunicaciones móviles y usuarios de redes fijas. Para poder acceder a la red, se necesita de una SIM (Módulo de identidad del abonado, por sus siglas en inglés), esta tarjeta contiene el número personal asignado al abonado y lo faculta para acceder a los servicios que haya concertado con la red GSM desde cualquier terminal (España, 2003).

2.2.4 Sensores, Actuadores y Transductores

Los sensores son dispositivos encargados de obtener información, es decir, de proporcionar señales de entrada a la unidad de control para que esta pueda determinar la orden de salida. Esta orden de salida es convertida en una señal eléctrica, la cual es enviada a un actuador que convertirá la energía eléctrica en otra forma de energía. Para simplificar se puede decir que el sensor envía información a la unidad de control, ésta la procesa y envía una orden, que recibe el actuador y se encarga de ejecutarla (Guarella, 2018).

2.2.4.1 Celda de Carga (20 Kg.)

Una celda de carga es un transductor que convierte una fuerza en una señal eléctrica, mediante una o más galgas internas, configuradas en un puente d Wheatstone. Existen diversos tipos de celdas de carga, la que se utiliza en este proyecto de tesis corresponde a 20 Kilogramos, que es el valor máximo que puede sensar (Naylamp, 2018). En la Figura 5 se aprecia la celda utilizada.

Figura 5: Celda de Carga de 20Kg.
Fuente: Elaboración propia

2.2.4.2 Transmisor Hx711

Es una interface entre la celda de carga y el microcontrolador, permite poder interpretar el peso de una manera sencilla. Internamente se encarga de la lectura del puente de Wheatstone, convirtiendo la lectura de analógica a digital con un conversor A/D de 24 bits. Además, posee una comunicación serial mediante 2 pines (Clock y Data) con el microcontrolador (Naylamp, 2018). En la Figura 6 se visualiza dicho transmisor.

2.2.4.3 Relé

Es un dispositivo de conmutación electrónico que realiza la conexión y desconexión de una carga con ausencia de contactos móviles en su interior. Por lo general, poseen 2 contactos, uno normalmente abierto y el otro normalmente cerrado. Se puede alimentar desde 3 a 5 voltios (Álvarez, 2010). En la Figura 7 se observa el relé utilizado.

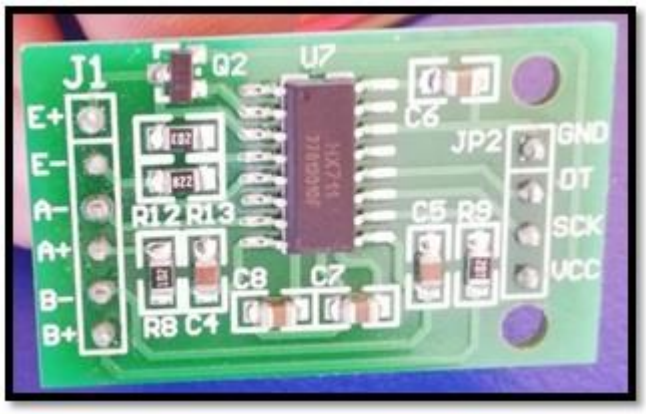


Figura 6: Hx711

Fuente: Elaboración propia

Figura 7: Relé

Fuente: Elaboración propia

2.2.4.4 Módulo GSM Sim900

Módulo que posee tecnología GMS/GPRS de la marca SIMCOM, se comunica con el microcontrolador a través de una interfaz serial y comandos AT. Soporta las 4 bandas de frecuencias internacionales de GSM, lo cual provee su compatibilidad con la mayoría de los operadores de telefonía a nivel global. Posee un regulador de voltaje y una ranura para insertar la tarjeta SIM (Montesdeoca, 2016). Se muestra el dispositivo en la Figura 8.

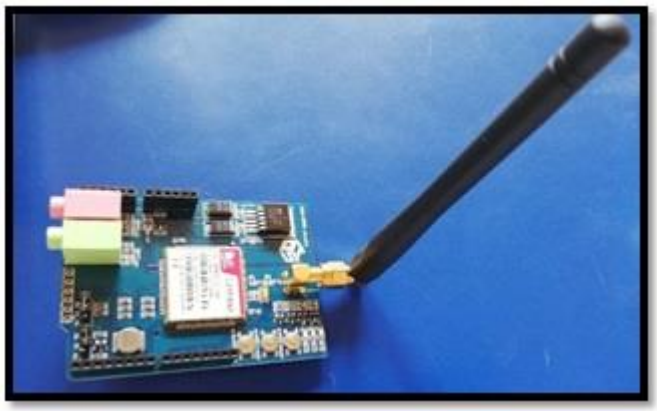


Figura 8: Módulo GSM SIM 900

Fuente: Elaboración propia

2.2.5 Microcontroladores

Es un circuito integrado que tiene la potestad de ejecutar órdenes, las cuales son grabadas en su memoria. Posee bloques funcionales que están diseñados para cumplir una tarea específica. Sus tres unidades funcionales son la unidad central de procesamiento, memoria y periféricos de entrada/salida (Valdés, 2007).

2.2.5.1 ATMEGA 328P

Es un Chip microcontrolador creado por ATMEL, de alto rendimiento con la capacidad de leer mientras escribe, 1KB de memoria EEPROM, 2KB de SRAM, 23 líneas de E/S, 32 registros de proceso general. Posee puerto serial, opera entre 1.8 y 5.5 voltios. Alcanza una respuesta promedio de 1 MIPS en un solo ciclo de reloj (Kurniawam, 2012).

2.2.6 Tarjetas Dedicadas

Sistema diseñado para realizar un propósito específico, la mayoría de sus componentes se encuentran incluidos en la placa base. Utilizan un procesador y una memoria relativamente pequeños. Su programación es directa al microcontrolador de la placa (Guerrero, 2014).

2.2.6.1 Raspberry Pi 3 Model B+

Raspberry Pi, si bien no es un sistema embebido propiamente definido, de acuerdo con la utilidad del presente proyecto, será definido de esta manera. La Raspberry Pi 3 B+ se fabricó por primera vez en marzo del 2018 para actualizar el modelo anterior la Raspberry Pi 3 Modelo B, posee mejoras, como lo es el nuevo procesador y cuenta con una mejor conectividad,

pasando de tener 1.2Ghz a tener 1.4Ghz, y en cuanto a la conectividad inalámbrica posee incorporado doble banda a 2.4GHz y 5GHz, un puerto Ethernet a 300 Mbits/s respecto de los 100 Mbits/s del modelo anterior. Además, ofrece conectividad Bluetooth 4.2 (Upton, 2014).

2.2.6.2 Cámara PI NOIR

Periférico que fabrica Raspberry para otorgarle captura de imágenes a la Raspberry Pi con 5 megapíxeles, pero que carece del filtro infrarrojo en el sensor. Por este motivo se ha convertido en la herramienta perfecta para la captura de imagen en condiciones muy pobres de luz y a un precio realmente bajo. Capaz de tomar fotos fijas con una resolución de 2592 x 1944, además de grabar vídeos en alta definición de hasta 1080p a 30 cuadros por segundo (Adén, 2014).

2.2.7 VNC Viewer

VNC es un sistema de cliente ultradelgado basado en un protocolo de pantalla simple que es independiente de la plataforma. Permite acceder a un solo escritorio desde distintos lugares simultáneamente, lo que hace posible compartir aplicaciones con un estilo de trabajo cooperativo y soportado por computadora (CSCW). La tecnología subyacente de VNC es un protocolo de visualización remota simple. Es la simplicidad de este protocolo lo que hace que VNC sea tan poderoso. Es un Software gratuito. (Richardson, 2008). En la Figura 9 se aprecia el entorno de VNC Viewer, donde se muestra ejemplos de conexiones previas con otros dispositivos, ya que el único requisito es conocer la dirección de red del equipo al que se desea conectar.

2.2.8 PuTTY

Es un cliente de SSH capaz de conectarse a servidores remotos iniciando una sesión en ellos. Posee soporte para conexiones de puerto serie local y por su origen en inglés, significa Puerto Único de Tipo Terminal. (Gálvez, 2015). La Figura 10 muestra la pantalla principal del programa.

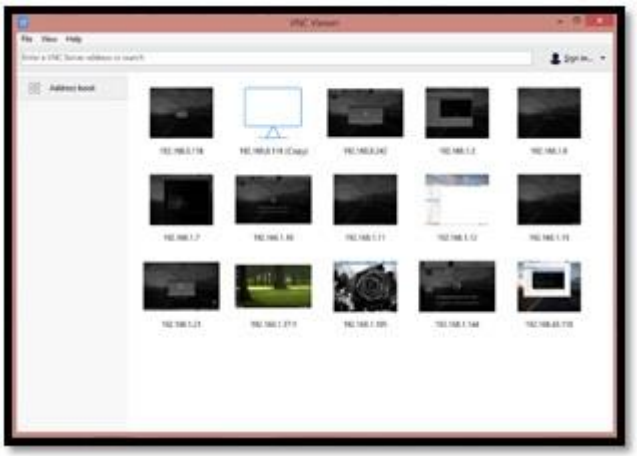


Figura 9: Pantalla de inicio de VNC Viewer

Fuente: Elaboración propia

Figura 10: Pantalla de inicio de PuTTY

Fuente: Elaboración propia

2.2.9 Open CV

Es una biblioteca libre destinada a la visión artificial desarrollada por Intel. Es multiplataforma, conteniendo en ella más de 500 funciones dentro de la gama de área del procesamiento de visión artificial (Bradskyi, 2015). Es de vital importancia para la elaboración de programas enfocados al procesamiento de imágenes.

2.2.10 Arduino

El software de código abierto facilita el poder realizar la escritura de códigos para poder grabarlos en el microprocesador. Es posible ejecutarlo en Windows, Mac OS y Linux. Se puede utilizar con cualquier placa Arduino. (Barret, 2010). La Figura 11 muestra el entorno de programación del software.

Figura 11: Pantalla de inicio de Arduino

Fuente: Elaboración propia

2.2.11 Normativas y Regulaciones

2.2.11.1 Ley N 30309:

Ley que promueve la investigación científica e innovación tecnológica. Publicada el 13 de marzo de 2015. Esta ley es necesaria para poder realizar investigaciones y aportes en el avance tecnológico del país, bajo una rigurosa supervisión de entidades públicas (L30309, 2015).

2.2.11.2 Recomendación UIT-R M.1073-1:

Establece recomendaciones sobre las características técnicas y de explotación de los sistemas celulares digitales de telecomunicaciones móviles terrestres para uso internacional y regional. Mediante la recopilación y comparación de las características de estos, así como la provisión de las referencias asociadas, la Recomendación suministra a las administraciones directrices para la evaluación de distintos sistemas celulares en sus aplicaciones proyectadas (UIT, 2012).

CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO

En el presente capítulo se brinda el detalle sobre el diseño e implementación del prototipo para el monitoreo de la posición de un bebé en su cuna. Inicialmente se realiza un estudio previo de las técnicas necesarias para la implementación y desarrollo del procesamiento de imagen; posteriormente, se describe el funcionamiento del hardware, el algoritmo desarrollado utilizando OpenCV en el Raspberry y finalmente la integración total de las partes.

3.1 Propuesta desarrollada

El desarrollo del prototipo se divide en lo concerniente a Hardware y Software:
Hardware: Consta de la implementación de la estructura física y esquemática de los componentes a utilizar, como la de la celda de peso y el módulo GSM Sim900. Software: Correspondiente al desarrollo del algoritmo para el procesamiento de imágenes con OpenCV y a la programación correspondiente para poder configurar y comunicar el Raspberry Pi con el ATMEGA 328P.

3.1.1 Electrónica del Prototipo

Para el desarrollo de las conexiones del Hardware y posterior integración de los componentes, es necesario conocer cuáles son las funciones de sus terminales. Para ello, se ahonda en el funcionamiento propio de cada elemento que se utiliza.

3.1.2 Relé

El relé trabaja con una lógica negada, de tal manera que al recibir 5V, se mantiene en el estado inicial. Cuando recibe 0V es cuando se realiza el cambio de estado del interruptor, físicamente hablando, y se activa la bobina del relé. Propiamente, el relé posee de 6 terminales de conexiones. Se activa o desactiva el pin digital de salida del ATMEGA 328p para su activación o desactivación. Además, el relé es utilizado para activar el funcionamiento del juguete distractor, el cual está temporizado para funcionar 3 minutos continuamente. En la Figura 12 se aprecian los pines de alimentación y control del relé.

Figura 12: Alimentación y Control del Relé

Fuente: Elaboración propia

Dependiendo de cómo se desee configurar la conexión, se posee un estado inicial de la bornera como normalmente abierta o normalmente cerrada. Entonces, tal como su nombre indica permite cerrar el circuito o no. Para el desarrollo del presente prototipo se utilizaron las borneras del estado Normalmente Abierto, debido a que al activarse el interruptor del relé por medio de la señal IN (señal de control), se procedió a cerrar el circuito, lo cual permitió que el juguete distractor de la cuna se active para entretener al bebé hasta la llegada de sus padres. En la Figura 13 se observan los pines mencionados.

Figura 13: Borneras de Conexión del Relé

Fuente: Elaboración propia

3.1.3 GSM SIM 900

Antes de empezar con la descripción de qué pines del módulo se utilizan, es necesario recordar que se le debe de insertar un chip móvil de cualquier operador. Asimismo, este módulo cuenta con una ranura en la parte posterior del mismo. Y,

para poder saber si es que la red ya se encuentra disponible al momento de energizarlo, una luz azul parpadeará de manera rápida aproximadamente por 5 segundos; este parpadeo disminuirá su frecuencia si es que ya se ha conectado a la red móvil. Además, este dispositivo tiene como alimentación 5V por medio del cable de color blanco y por el cable de color negro se presenta la tierra (GND). Por otro lado, la comunicación Serial Tx/Rx se realiza por medio del pin 02 y 03 del módulo. Y, el pin 08 se utiliza para comandar la activación del Relé. En la Figura 14 se aprecia lo mencionado.

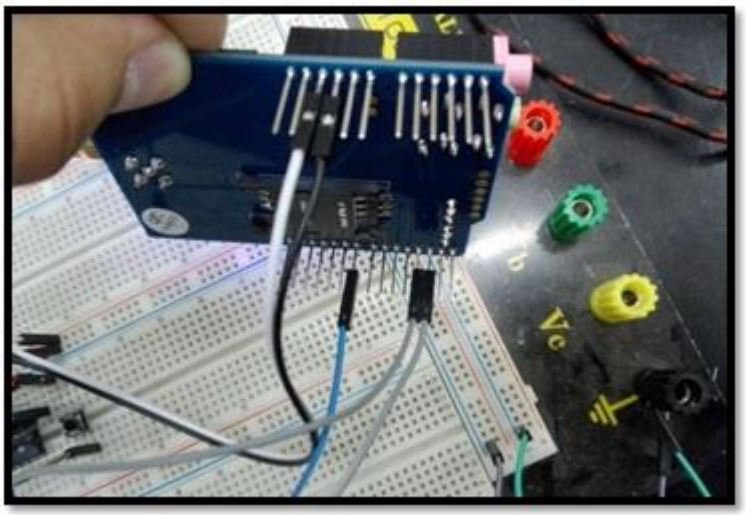


Figura 14: Diagrama del GSM SIM 900

Fuente: Elaboración propia

3.1.4 Celda de peso

La celda de peso a utilizar posee una capacidad de 20 kilogramos como máximo, previo a su utilización se tuvo que realizar una calibración del "0" y posteriormente de la tara, la cual corresponde a la estructura montada sobre la celda y está compuesta por una plancha de madera montada sobre la propia celda y el colchón de la cuna. El principio de funcionamiento de la celda se basa en la deformación de su masa, variando así su resistencia; por ello, cuando el bebé se posicione sobre la celda, permite conocer su posición cercana a la baranda, para lo cual se emite una alerta a los padres. La variación de voltaje, que es la salida de la celda, se dirige hacia los terminales del Transductor Hx711, este componente posee un reloj síncrono que posibilita la transmisión de datos con el ATMEGA 328P, para obtener el peso actual medido. En la Figura 15, se observa el diagrama de conexiones de la celda de pesaje, donde el terminal 1 es la alimentación de 5V,

el terminal 2 y 3 corresponden a la lectura de variación del voltaje de la celda; finalmente, el terminal 4 es la tierra.

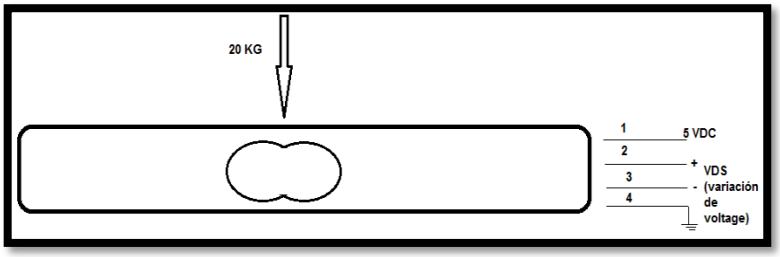


Figura 15: Diagrama de Celda de Peso

Fuente: Elaboración propia

En la Figura 16, se visualiza el diagrama de conexiones del transductor Hx711, en donde el sentido de la señal es de izquierda a derecha. Luego, el pin "+E" se conecta con el terminal 1 proveniente de la celda; el pin "-E" se conecta con el terminal 4 proveniente de la celda; el pin "+A" se conecta con el terminal 3 proveniente de la celda; por último, el pin "-A" se conecta con el terminal 4 de la celda. Los terminales de salida "SCK" (Reloj síncrono) y "DT" (Transmisión de datos) se conectan con el ATMEGA 328P en los pines A0 y A1 que corresponden a entradas análogas, respectivamente.

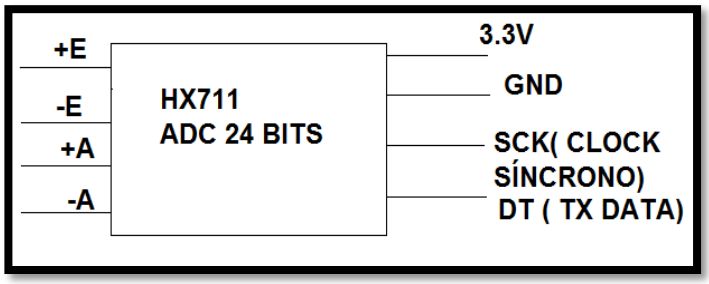


Figura 16: Diagrama de Transductor Hx711

Fuente: Elaboración propia

Como ya es conocida la función de cada terminal, se procede a integrar los componentes, siendo el diagrama circuital correspondiente a la Tabla 1, en la cual se aprecia la asignación de pines al ATMEGA 328P.

Tabla 1: Asignación de pines del ATMEGA 328p

ATMEGA 328p						
Pines	**ATMEGA328P**	**Función**	**Balanza1**	**Balanza2**	**SIM900**	**RELÉ**
1	RST	Reset				
2	D0/RX	Serial rx				
3	D1/TX	Serial tx				
4	D2	GPIO2			PIN2	
5	D3	GPIO3			PIN3	
6	D4	GPIO4				IN
7	VCC	5VDC	E+	E+		VCC
8	GND	Gnd	E-	E-		GND
9	XTAL1	Crystal				
10	XTAL2	Crystal				
11	D5	GPIO5				
12	D6	GPIO6				
13	D7	GPIO7				
14	B0	GPIO8			PIN8	
15	B1	GPIO9				
16	B2	GPIO10				
17	B3	GPIO11				
18	B4	GPIO12				
19	B5	GPIO13				
20	AREF+	5VDC	VCC	VCC	VCC	
21						
22	AREF-	Gnd	GND	GND	GND	
23	C0	Analog0	CLK			
24	C1	Analog1	DT			
25	C2	Analog2		CLK		
26	C3	Analog3		DT		
27	C4	Analog4				
28	C5	Analog5				

Fuente: Elaboración propia

3.1.5 Software del Prototipo

En cuanto al software del prototipo, se procedió al desarrollo de la programación específica de cada una de las partes y su posterior integración. Dentro de este desarrollo, se muestra la programación individual e integrada de los elementos que componen el prototipo, enfocándose tanto en el ATMEGA 328P que controla a los actuadores y la comunicación serial entre el ATMEGA 328P y el Raspberry Pi 3 modelo B+.

3.1.5.1 Programación de ATMEGA 328p

Para la programación se asignó la salida digital que controla la activación del relé correspondiente al PIN N°6 del ATMEGA 328p. Para el Módulo GSM Sim900 se le asignan los PIN's N° 4, 5 y 14 tal como lo indica la

tabla 1; y, para las celdas de peso los PIN's N° 23, 24, 25 y 26, respectivamente. En la programación se adhiere el comando que proporciona la comunicación serial entre el Raspberry y el ATMEGA 328p. La lógica de activación de los actuadores corresponde al comando "IF" y al valor del carácter recibido; de esta manera, de acuerdo al resultado del procesamiento de imagen, se configuró la programación para que el Raspberry envíe un carácter de texto simbolizado por una letra del abecedario, de tal forma que active la emisión de la alerta por parte del módulo GSM Sim900. Asimismo, la activación del juguete de cuna fue de la mano con la emisión de la alerta. En la Figura 17 se aprecia el diagrama de bloques del programa.

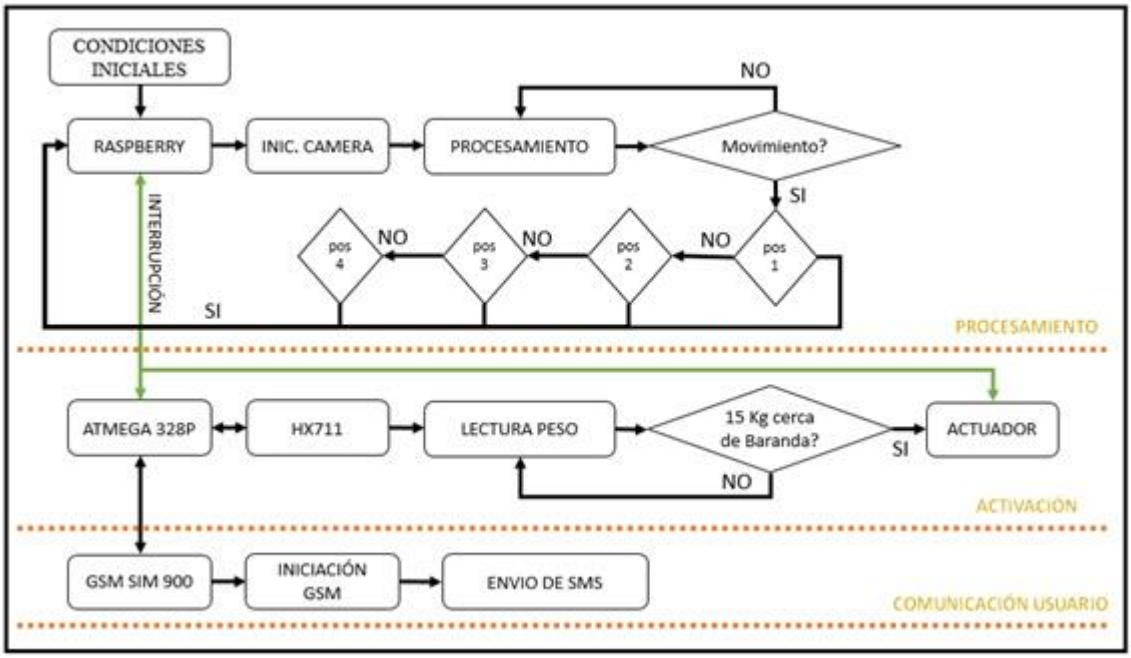


Figura 17: Diagrama de Bloques del Programa

Fuente: Elaboración propia

3.1.5.2 Programación de Raspberry Pi 3 modelo B+

La programación del Script que controló el procesamiento de imagen en el Raspberry Pi, se encuentra en el Anexo 1. Ahí se aprecia la secuencia de pasos utilizados para establecer el resultado entre las 4 posiciones del bebé con la función de "Diferencia Normalizada", y el algoritmo del Template Matching. Además, para distinguir los resultados de cada posición, se realizó una comparación de cada resultado, hallando el máximo valor

localizado y el mínimo valor localizado. Una vez conocida la posición del bebé, se procedió a enviar el carácter por medio de una comunicación serial entre ambos dispositivos, momento en el cual, se otorgó el funcionamiento al ATMEGA 328P para la activación de los actuadores y el mensaje a enviar por medio del GSM Sim900.

3.1.6 Programación del Algoritmo para la Visión Artificial

El presente proyecto de tesis contó con seis funciones distintas que fueron empleados con el algoritmo de procesamiento de imagen de la técnica de Template Matching; de las cuales se procedió a escoger solo una, a través de pruebas, con la finalidad de corroborar su eficacia. Las funciones son las siguientes:

- Suma de Diferencias Cuadradas o SSD

  Esta función también es conocida como Distancia Euclidiana simple, pero al cuadrado. Y consiste en tomar cada par de píxeles y restarlos, posteriormente sumando todos los cuadrados de cada resultado.

  CV_TM_SQDIFF

  $$R(x,y) = \sum_{x',y'}(T(x',y') - I(x+x',y+y'))^2 \quad (1)$$

  Donde:

  T: Corresponde a la imagen Template o muestra que se tiene que encontrar o comparar con la imagen en tiempo real del bebé.

  I: Corresponde a la imagen capturada en tiempo real de la posición del bebé.

  (x,y): Son las coordenadas de un pixel de la imagen capturada. En la ecuación se coloca el valor del píxel en esa ubicación.

  (x',y'): Son las coordenadas de un pixel de la imagen Template o muestra. En la ecuación se coloca el valor del píxel en esa ubicación.

  R: Es el resultado obtenido de la sumatoria.

- Suma de Diferencias Cuadradas Normalizada

  Es una función que no se utiliza mucho en la práctica, pues la normalización se realiza con el denominador que se aprecia y es el mismo para todas las demás funciones normalizadas.

  CV_TM_SQDIFF_NORMED

  $$R(x,y) = \frac{\sum_{x',y'}(T(x',y') - I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \quad (2)$$

Este factor es la suma de la imagen plantilla o Template al cuadrado, con la "ventana" de la imagen al cuadrado. La "ventana" es el valor del pixel resultante de la suma de las coordenadas de ambas imágenes (Template y captura) al cuadrado. Las variables son las mismas que en la función anterior y que en las siguientes, por lo que se omite redundar en una misma explicación.

- Correlación Cruzada

  Básicamente es un producto de punto, toma cada par de píxeles y multiplica, para luego proceder a sumar todos los productos.

  CV_TM_CCORR

  $$R(x,y) = \sum_{x',y'}(T(x',y').I(x+x',y+y')) \quad (3)$$

- Correlación Cruzada Normalizada

  Tiene el mismo proceder que la Correlación cruzada, pero tal cual se aprecia en su fórmula se divide por la "ventana" de la imagen.

  CV_TM_CCORR_NORMED

  $$R(x,y) = \frac{\sum_{x',y'}(T(x',y').I(x+x',y+y'))}{\sqrt{\sum_{x',y'}T(x',y')^2.\sum_{x',y'}I(x+x',y+y')^2}} \quad (4)$$

- Coeficiente Cruzado

  Similar a la Correlación Cruzada, pero se encuentra normalizada con sus propias covarianzas. Cabe recordar que la covarianza es un valor indicativo del grado de variación entre dos variables aleatorias respecto a sus medias. Este dato sirve para poder hallar el coeficiente de correlación lineal y estimar la dependencia entre ambas variables.

  CV_TM_CCOEFF

  $$R(x,y) = \sum_{x',y'}(T'(x',y').I(x+x',y+y')) \quad (5)$$

  Donde:

  $$T'(x',y') = T(x',y') - \frac{1}{w.h}.\sum_{x'',y''}T(x'',y'')$$

  $$I'(x+x',y+y') = I(x+x',y+y') - \frac{1}{w.h}.\sum_{x'',y''}I(x+x'',y+y'')$$

  W, H: Son los valores del ancho y alto de la imagen Template respectivamente.

  T': Corresponde a la covarianza de los valores de la imagen Template.

  I': Corresponde a la covarianza de los valores de la imagen capturada en tiempo real.

- Coeficiente Cruzado Normalizado

  Se rige bajo la misma lógica que el Coeficiente Cruzado, salvo que se encuentra normalizado.

  CV_TM_CCOEFF_NORMED

$$R(x,y) = \frac{\sum_{x\prime,y\prime}(T\prime(x',y').I\prime(x+x',y+y'))}{\sqrt{\sum_{x\prime,y\prime}T(x',y')^2.\sum_{x\prime,y\prime}I(x+x',y+y')^2}} \quad (6)$$

3.1.7 Explicación del Algoritmo de programación

A continuación, se explica a detalle la programación empleada para el procesamiento de imagen que detecta la posición del bebé en su cuna.

En la Figura 18 se muestran las librerías que se han utilizado para la realización de las pruebas y el adecuado funcionamiento del algoritmo "Template Matching". Pues sin ellas, muchos de los comandos y funciones utilizados no podrían ser ejecutados.

```
#!/usr/bin/env python
import os
import time
import picamera
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

Figura 18: Librerías para el Procesamiento de Imagen con Open CV

Fuente: Elaboración propia

La Figura 19 muestra la secuencia de encendido de la cámara PiNoir de Raspberry para la captura de la posición actual del bebé en su cuna. Se aprecia en la Figura 20 que la foto se guarda con el nombre "fotoreal.jpg" y por fines de pruebas el retardo de la captura es de 10 segundos para poder acomodar al muñeco que representa a un bebé de acuerdo a lo necesitado. La imagen se guarda en la carpeta local de trabajo del Raspberry Pi.

```
with picamera.PiCamera() as picx:
    picx.start_preview()
    time.sleep(10)
    picx.capture('fotoreal.jpg')
    picx.stop_preview()
    picx.close()
```

Figura 19: Captura de Imagen del Bebé en su Cuna

Fuente: Elaboración propia

```
time.sleep(3)
img = cv2.imread('fotoreal.jpg',0)
img2 = img.copy()
template = cv2.imread('bocarriba2.png',0)
w, h = template.shape[::-1]
```

Figura 20: Comando template shape

Fuente: Elaboración propia

Asimismo, se aprecia que existe un retardo de 3 segundos antes de proceder a leer la imagen capturada en pantalla, para luego continuar haciendo una copia de la misma. La imagen que se lee como la variable "template" es la posición que se tiene como base de datos, la cual se compara con la imagen capturada en el algoritmo para corroborar la detección de la posición del bebé. El comando "template.shape" se utiliza para conocer las coordenadas de Ancho (w por su escritura en inglés) y Alto (h por su escritura en inglés) de la imagen.

Asimismo, se tiene como matriz "methods" las seis funciones sometidas a prueba: se escoge solo una para el algoritmo final, porque se debe de establecer un patrón de resultados para realizar el discernimiento de las posiciones; consecuentemente, no es necesario realizar el procesamiento con cada método porque es perder tiempo y memoria con resultados redundantes y/o similares. La Figura 21 muestra la matriz de funciones. Además, al momento de ejecutar el Script, se evalúan todas las funciones de la matriz "methods" de manera secuencial.

```
methods = ['cv2.TM_CCOEFF', 'cv2.TM_CCOEFF_NORMED', 'cv2.TM_CCORR',
        'cv2.TM_CCORR_NORMED', 'cv2.TM_SQDIFF', 'cv2.TM_SQDIFF_NORMED']
for meth in methods:
    img = img2.copy()
    method = eval(meth)
```

Figura 21: Funciones de Prueba

Fuente: Elaboración propia

La variable "res" almacena el resultado del "Matching" donde se evalúa la imagen "template" dentro de la imagen "img" con la función correspondiente. Luego, para conocer los valores necesarios del resultado, se tienen las variables "min_val" que es el mínimo valor del resultado con la función utilizada, "max_val" que es el máximo valor del resultado con la función utilizada, "loc_min" que es el mínimo valor de la coordenada de posición del resultado encontrado con la función utilizada, y "loc_max" que es el máximo valor de la coordenada de posición del resultado encontrado con la función utilizada. En la Figura 22, se puede apreciar que utilizando el comando "cv2.minMaxLoc()" se obtuvo el valor de las variables mencionadas.

```
res = cv2.matchTemplate(img,template,method)

min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res)
```

Figura 22: Variables que Almacenan Resultados

Fuente: Elaboración propia

De acuerdo con las ecuaciones de cada función mostrada anteriormente, las únicas que utilizan la diferencia de valores son CV_TM_SQDIFF y CV_TM_SQDIFF_NORMED, a diferencia de los 4 restantes que multiplican los valores. Es por eso que para localizar cuánto concuerda la vecindad de ese píxel con el de la plantilla, para los dos métodos mencionados, se toma la menor localización. Por otro lado, para los métodos restantes se toma la mayor. La Figura 23 reseña lo mencionado. Luego de ello, la variable "bottom_right" almacena la suma de las coordenadas anteriormente almacenadas (w,h) con la ubicación

encontrada que se almacena con el nombre de “top_left”. El cálculo de “bottom_right” es la suma desde el punto o píxel con más similitud a la plantilla con el largo y alto de la misma.

```
if method in [cv2.TM_SQDIFF, cv2.TM_SQDIFF_NORMED]:
  top_left = min_loc
else:
  top_left = max_loc
bottom_right = (top_left[0] + w, top_left[1] + h)
```

Figura 23: Localización del Mínimo y Máximo Valor de Coordenada

Fuente: Elaboración propia

En la Figura 24 el círculo representa la vecindad exacta o más parecida a la plantilla, por lo que desde el píxel con mayor similitud nace el rectángulo indicando cuál es el área correspondiente al resultado. Desde el punto de vista visual, es la porción de la imagen correspondiente a la plantilla.

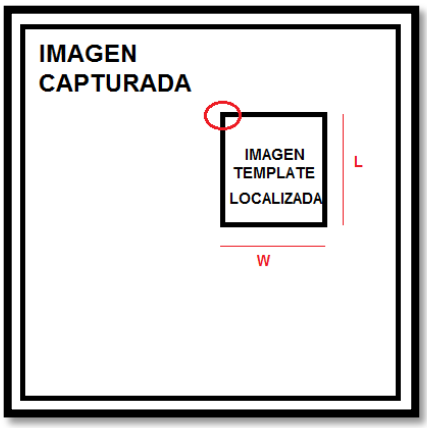


Figura 24: Asignación de la Delimitación del Resultado

Fuente: Elaboración propia

Finalmente, con fines de conocer los resultados encontrados, se imprimen en la pantalla los valores de cada variable; posterior a ello, se procede a la representación gráfica de la imagen con el resultado del emparejamiento y con la imagen de cómo se ha determinado la correspondiente porción. En la Figura 25 se aprecia lo mencionado.

```
print min_val
print max_val
print  min_loc
print max_loc
cv2.rectangle(img,top_left, bottom_right, 255, 2)
plt.subplot(121),plt.imshow(res,cmap = 'gray')
plt.title('Matching Result'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(img,cmap = 'gray')
plt.title('Detected Point'), plt.xticks([]), plt.yticks([])
plt.suptitle(meth)


plt.show()
```

Figura 25: Visualización de Resultados

Fuente: Elaboración propia

A continuación, se presentan las imágenes obtenidas luego de haber utilizado los seis métodos con dos imágenes distintas como "Template", las cuales corresponden a las Figuras 26 y 27.

Figura 26: Bebé "De Lado"

Fuente: Elaboración propia

Figura 27: Bebé “caído”
Fuente: Elaboración propia

De la misma manera, las Figuras 28 y 29 que se muestran a continuación, son las capturas que corresponden a la posición actual del bebé en su cuna, las cuales se encuentran almacenadas en la memoria como “fotoreal.jpg”.

Figura 28: Bebé en la Cuna en posición “Caído”
Fuente: Elaboración propia

Figura 29: Bebé en la Cuna en posición “De Lado”

Fuente: Elaboración propia

Asimismo, desde la Figura 30 a la Figura 47, se muestran los resultados de lo mencionado. Además, los resultados numéricos mostrados se encuentran ordenados de la siguiente manera: “min_loc”, “max_loc”, “min_val” y “max_val”, y solo se incluyen cinco funciones porque ya se ha seleccionado una. Los primeros 5 resultados mostrados son en base a la plantilla que muestra al bebé de lado; y los otros 5 resultados son en base a la plantilla que muestra al bebé “caído” en su cuna. Para aclarar, la imagen de la izquierda del resultado ilustrado corresponde visualmente a cómo se ve la captura al momento de ser ejecutado el comando “cv2.matchTemplate()”, ya que se le ha dado un tratamiento previo antes de la comparación con la imagen base o Template.

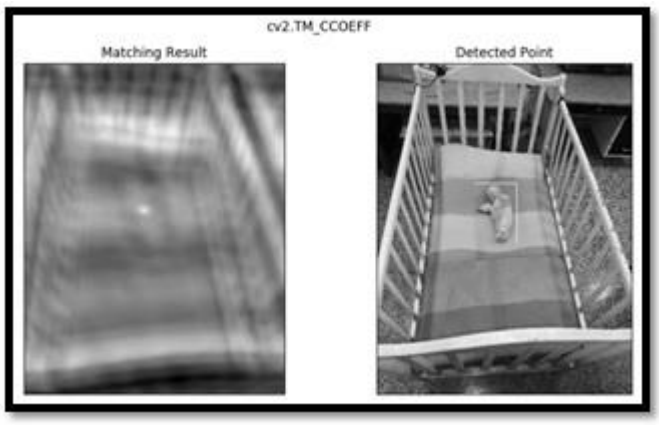


Figura 30: Resultado Ilustrado de función “cv2.TM_CCDEFF”

Fuente: Elaboración propia

Figura 31: Resultado Numérico de función “cv2.TM_CCDEFF”

Fuente: Elaboración propia

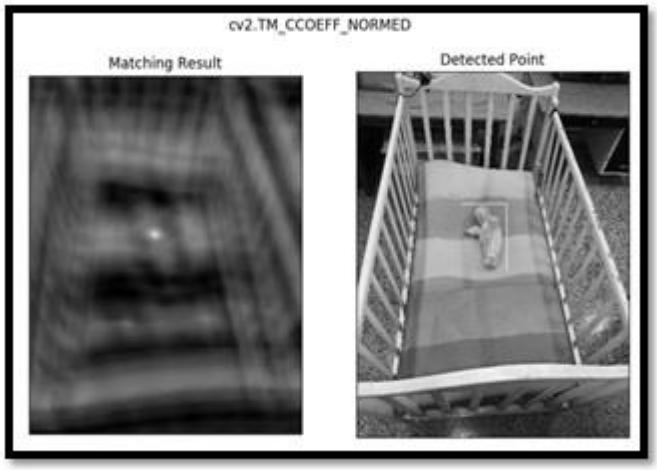


Figura 32: Resultado Ilustrado de función “cv2.TM_CCDEFF_NORMED”

Fuente: Elaboración propia

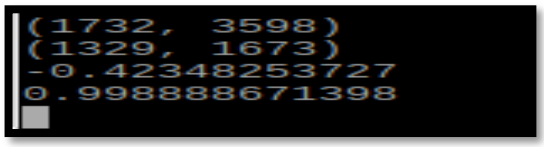


Figura 33: Resultado Numérico de función “cv2.TM_CCDEFF_NORMED”

Fuente: Elaboración propia

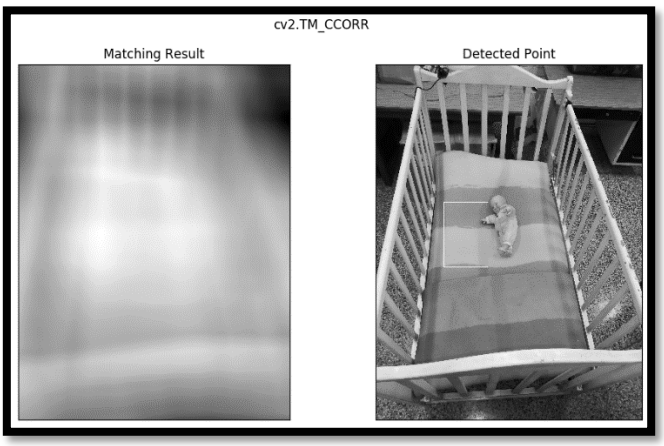


Figura 34: Resultado Ilustrado de función “cv2.TM_CCORR”

Fuente: Elaboración propia

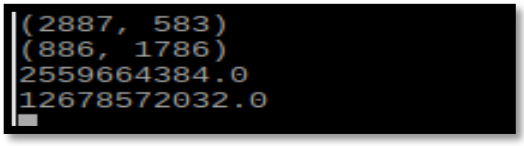


Figura 35: Resultado Numérico de función “cv2.TM_CCORR”

Fuente: Elaboración propia

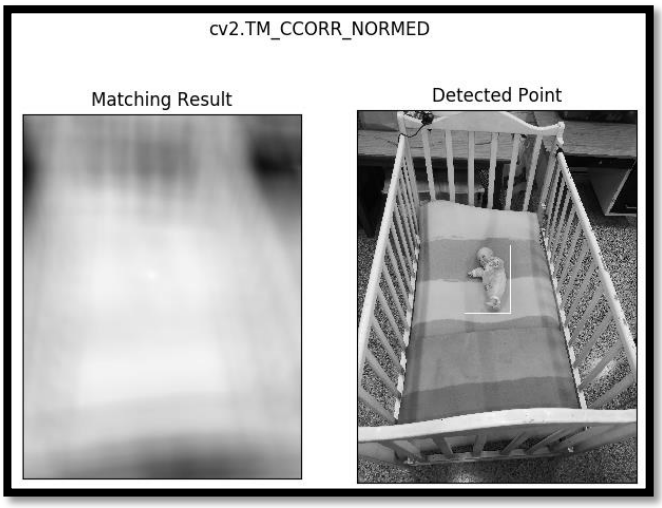


Figura 36: Resultado Ilustrado de función “cv2.TM_CCORR_NORMED”

Fuente: Elaboración propia

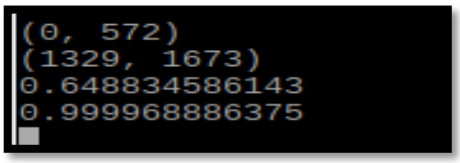


Figura 37: Resultado Numérico de función “cv2.TM_CCORR_NORMED”

Fuente: Elaboración propia

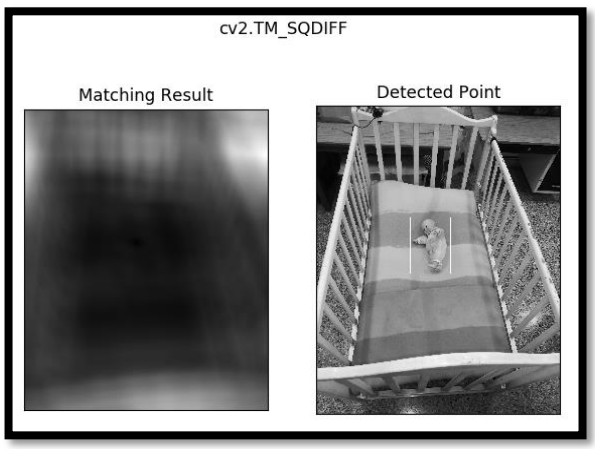


Figura N° 38: Resultado Ilustrado de función “cv2.TM_SQDIFF”

Fuente: Propia

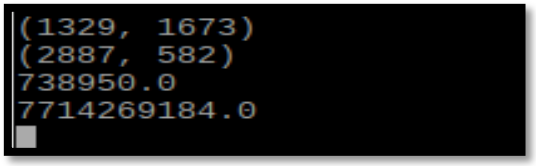


Figura 39: Resultado Numérico de función “cv2.TM_SQDIFF”

Fuente: Elaboración propia

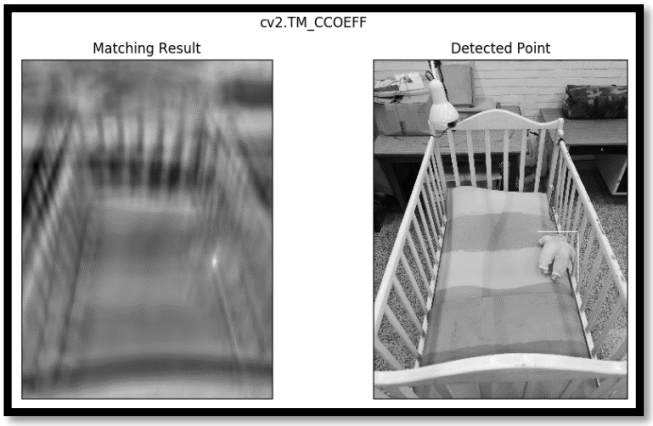


Figura 40: Resultado Ilustrado de función “cv2.TM_CCOEFF”

Fuente: Elaboración propia

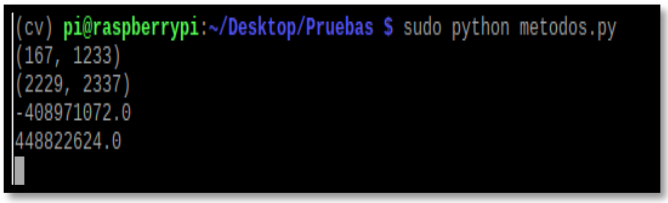


Figura 41: Resultado Numérico de función “cv2.TM_CCOEFF”

Fuente: Elaboración propia

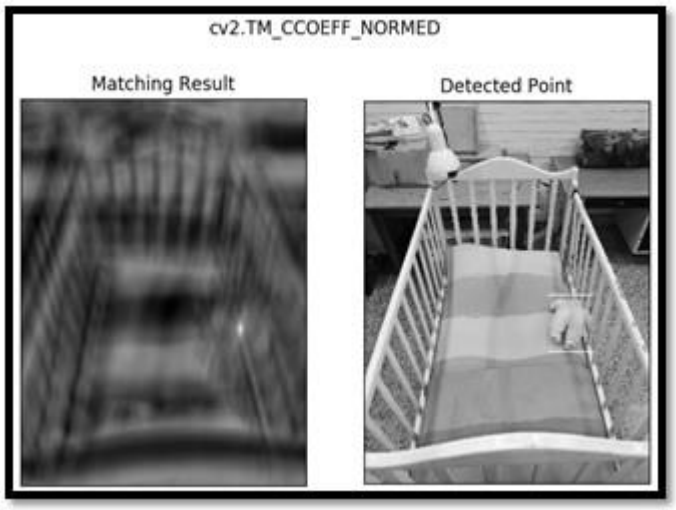


Figura 42: Resultado Ilustrado de función “cv2.TM_CCOEFF_NORMED”

Fuente: Elaboración propia

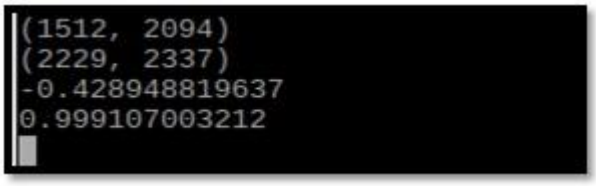


Figura 43: Resultado Numérico de función “cv2.TM_CCOEFF_NORMED”

Fuente: Elaboración propia

Figura 44: Resultado Ilustrado de función “cv2.TM_CCORR”

Fuente: Elaboración propia

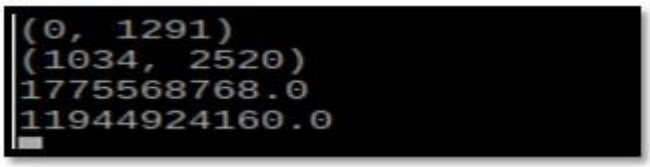


Figura 45: Resultado Numérico de función “cv2.TM_CCORR”

Fuente: Elaboración propia

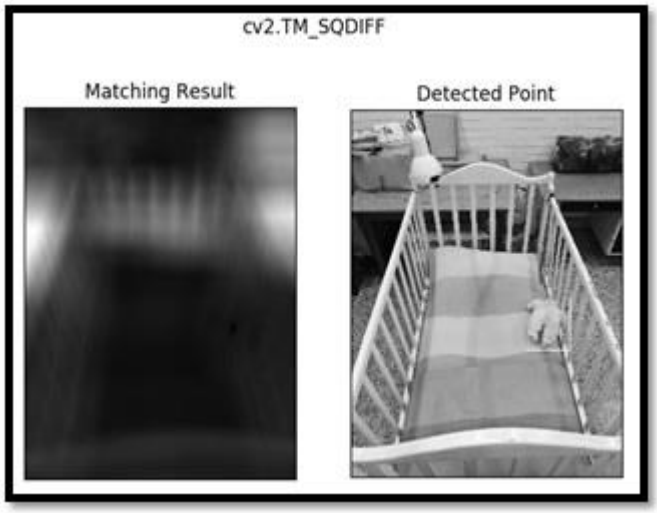


Figura 46: Resultado Ilustrado de función “cv2.TM_SQDIFF”

Fuente: Elaboración propia

Figura 47: Resultado Numérico de función “cv2.TM_SQDIFF”

Fuente: Elaboración propia

Así como también, el resultado numérico indica las coordenadas del límite de detección; es decir, el rectángulo que se aprecia en el resultado ilustrado. De esta manera, tales coordenadas señalan el vértice superior izquierdo y el inferior derecho, respectivamente. Luego, se aprecia debajo el valor mínimo y máximo, correspondientemente, del resultado con el método escogido.

Asimismo, se escoge la función CV.TM_SQDIFF_NORMED, ya que como se basa en la diferencia, está normalizada, y así como también se puede establecer una lógica comparativa para determinar en qué posición se encuentra el bebé. Por otro lado, se aprecia que en los resultados anteriores el único método que falla es el “CV.TM_CORR”. A continuación, se presentan los resultados de las pruebas realizadas con el método escogido y con todas las posiciones del bebé evaluado. Debido a que la realización de pruebas con un bebé real es arriesgado y complicado, se tomó la decisión de utilizar para todas las pruebas un “bebé muñeco”, por lo que a partir de este punto cuando se refiera a bebé, se debe entender que es un muñeco.

Los resultados apreciados desde la Figura 48 hasta la Figura 55 corresponden a las posiciones “boca arriba”, “boca abajo”, “de lado” y “caído”.

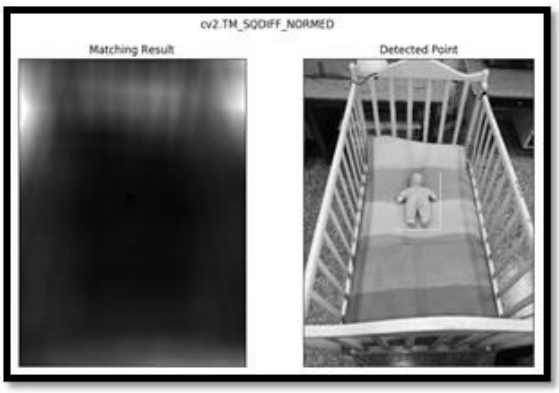


Figura 48: Resultado Ilustrado de función “cv2.TM_SQDIFF_NORMED” en posición “boca abajo”.

Fuente: Elaboración propia

Figura 49: Resultado Numérico de función “cv2.TM_SQDIFF_NORMED” en posición “boca abajo”.

Fuente: Elaboración propia

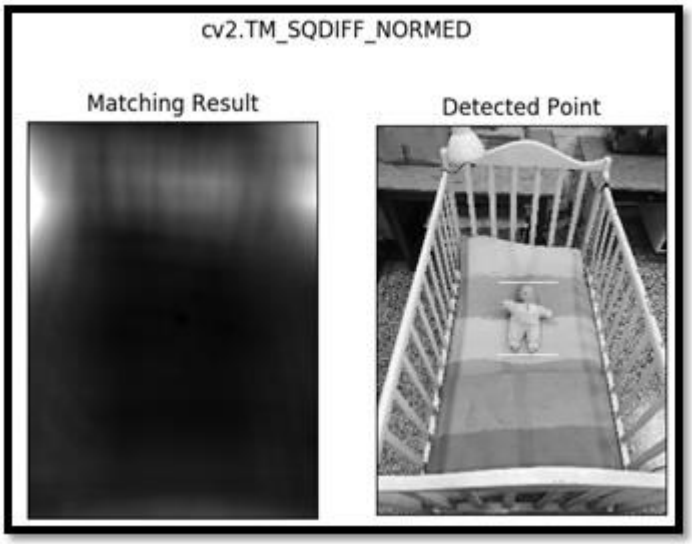


Figura 50: Resultado Ilustrado de función “cv2.TM_SQDIFF_NORMED” en posición “boca arriba”.

Fuente: Elaboración propia

Figura 51: Resultado Numérico de función “cv2.TM_SQDIFF_NORMED” en posición “boca arriba”.

Fuente: Elaboración propia

Figura 52: Resultado Ilustrado de función “cv2.TM_SQDIFF_NORMED” en posición “caído”.

Fuente: Elaboración propia

Figura 53: Resultado Numérico de función "cv2.TM_SQDIFF_NORMED" en posición "caído".
Fuente: Elaboración propia

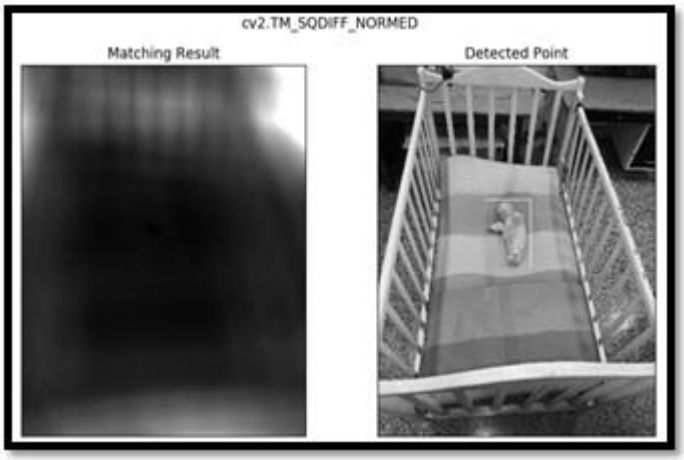


Figura 54: Resultado Ilustrado de función "cv2.TM_SQDIFF_NORMED" en posición "De lado".
Fuente: Elaboración propia

Figura 55: Resultado Numérico de función "cv2.TM_SQDIFF_NORMED" en posición "De lado".
Fuente: Elaboración propia

De las figuras anteriores, se aprecia que no hay falla en los resultados debido a que se detecta la ubicación correspondiente del bebé. Sin embargo, esta demostración solo es para indicar que la función es fiable, así como también el procesamiento con la técnica Template Matching; ya que el verdadero objetivo del presente proyecto de investigación es discernir en qué posición se encuentra el bebé en su cuna, por ende, la programación se expande e incrementa. Además, el nuevo programa posee las cuatro posiciones como plantillas y otorga el resultado de la posición actual del bebé en su cuna. Para poder llevar las pruebas de manera adecuada, se asignó un número a cada posición, tal como se indica en la Tabla 2.

Tabla 2: Número de Posición

Número de Posición	Estado del bebé en su cuna
1	De Lado
2	Boca Abajo
3	Tirado
4	Boca Arriba

Fuente: Elaboración propia

Desde la Figura 56 hasta la 63 se aprecia visualmente la detección de la posición en la que se encuentra el bebé luego del procesamiento de la imagen. Como ya ha sido mencionado, la imagen de la izquierda es la representación de cómo se ve la imagen que se ha capturado con la cámara, al momento previo de ser comparada con la imagen Template. El rectángulo que indica el lugar de la posición detectada, es decir al bebé en la cuna, no se aprecia completamente debido al tamaño de la pantalla en la que se ven los resultados, por lo cual no es un error.

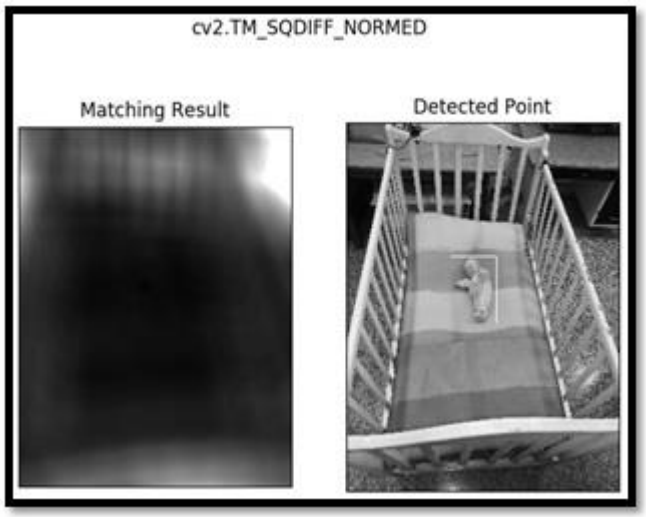


Figura 56: Resultado Ilustrado en la posición 1.

Fuente: Elaboración propia

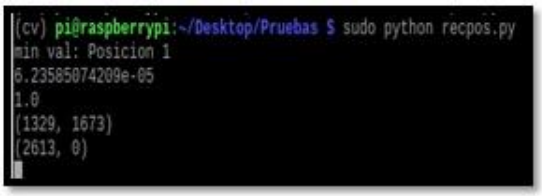


Figura 57: Resultado Numérico en la posición 1.

Fuente: Elaboración propia

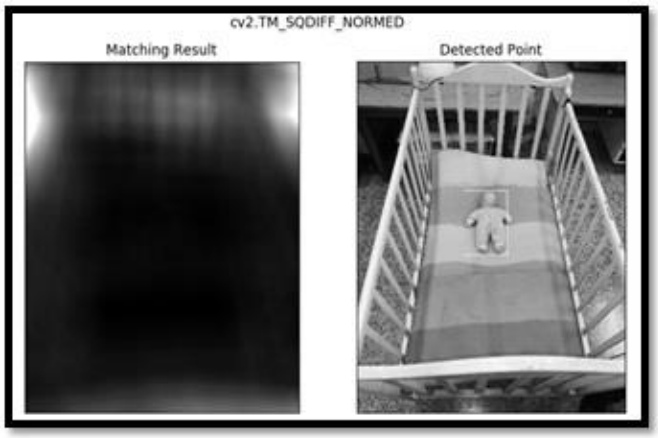


Figura 58: Resultado Ilustrado en la posición 2.

Fuente: Elaboración propia

Figura 59: Resultado Numérico en la posición 2.

Fuente: Elaboración propia

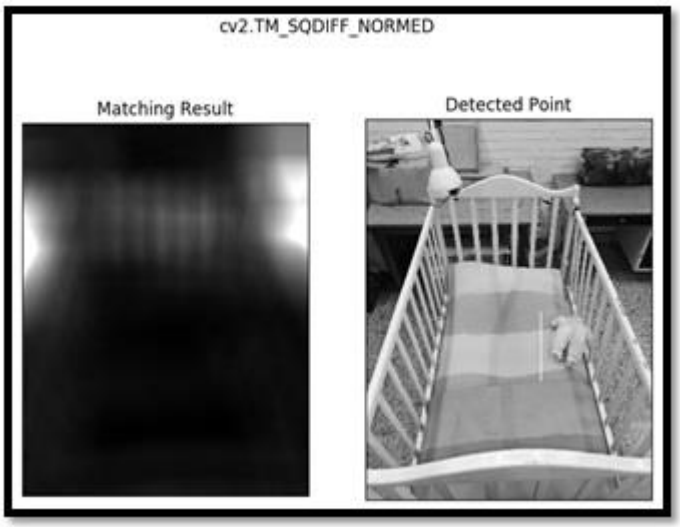


Figura 60: Resultado Ilustrado en la posición 3.

Fuente: Elaboración propia

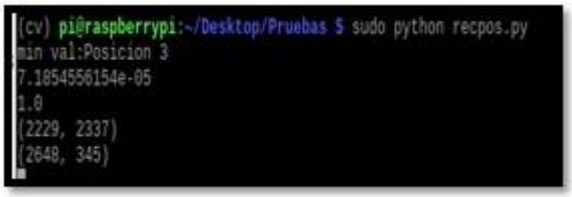


Figura 61: Resultado Numérico en la posición 3.

Fuente: Elaboración propia

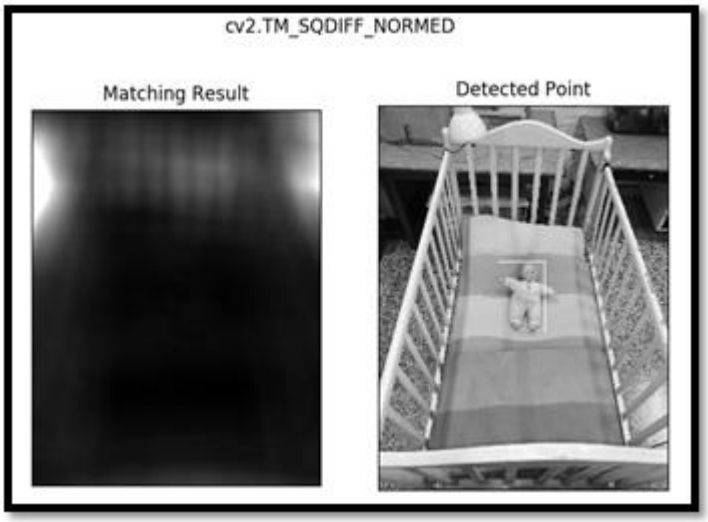


Figura 62: Resultado Ilustrado en la posición 4.

Fuente: Elaboración propia

Figura 63: Resultado Numérico en la posición 4.

Fuente: Elaboración propia

Igualmente, se observa que los resultados no presentan fallo y que se acierta en totalidad con la distinción de las posiciones, por ello que se procede a asignarle a cada posición un carácter que será enviado al ATMEGA 328P mediante comunicación serial, para que este active al módulo GSM SIM900 de acuerdo a la Tabla 3.

La asignación mencionada en la Tabla 3 permite ordenar y conocer el mensaje de texto que se debe de recibir, luego de obtener un resultado del procesamiento de imagen.

Tabla 3: Asignación de Caracteres a Posición de Bebé

Carácter	Posición	Mensaje
A	De Lado	----------
B	Boca Abajo	"El bebé está Boca abajo"
C	Caído	"El bebé se ha caído"
D	Boca Arriba	"El bebé está Boca arriba"

Fuente: Elaboración propia

3.2 Descripción del Prototipo

El controlador utilizado fue el Raspberry el cual hace uso del sistema ARM 1176JZF-S a 700 MHz, además está integrado con una tarjeta de Arduino controlada por el ATMEGA 328P. Ambos mantienen una comunicación serial UART la cual permite el control del actuador luminoso (juguete distractor de cuna) y de la mensajería (SIM900) que informa el estado del bebé. En el Anexo 1 se aprecia la asignación de pines de fábrica del Raspberry. Además, el diagrama de bloques se puede observar en la Figura 64. En cuanto al procesamiento de imagen se llevó a cabo en el Raspberry, permitiendo detectar y reconocer la posición del bebé en la cuna, e identificando los siguientes estados: De lado, boca arriba, boca abajo y caído en la cuna.

En cuanto al flujograma, se establece de la siguiente manera. Una vez identificada la posición del bebé, la comunicación entre el Raspberry y el ATMEGA 328P permitió activar un elemento que genera distracción, y por lo cual distrae al bebé. A su vez, se envía un mensaje de texto para alertar a los padres y/o responsables del cuidado para que acudan y atiendan al menor. La celda de pesaje aguarda a emitir una alerta en el momento que el bebé se ubique cerca de la baranda. En la Figura 65, se aprecia el diagrama con las señales electrónicas de voltajes y corrientes del prototipo, así como las etapas de funcionamiento del mismo.

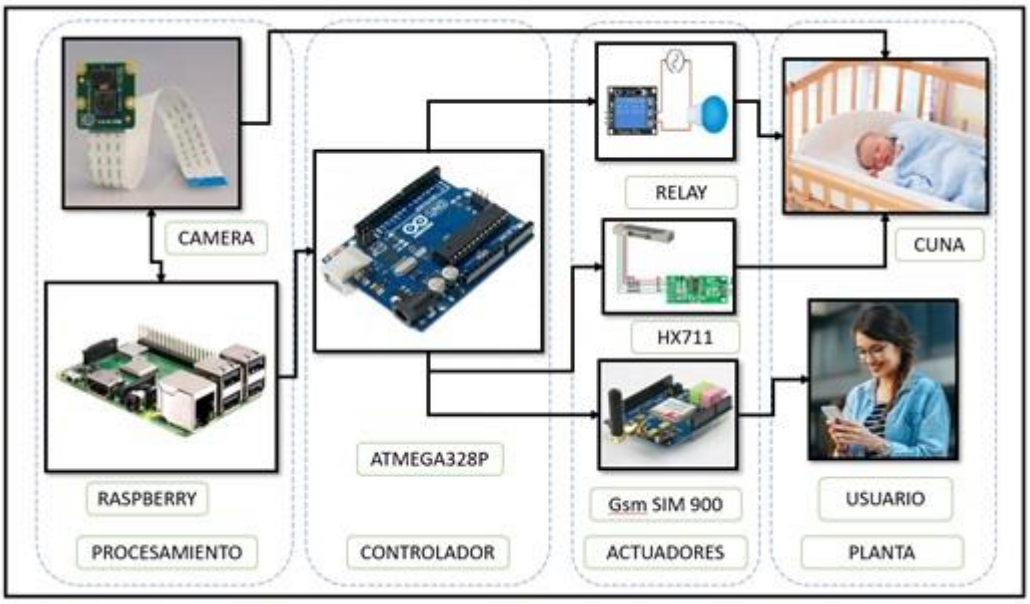


Figura 64: Diagrama de bloques de conexión del Prototipo

Fuente: Elaboración propia

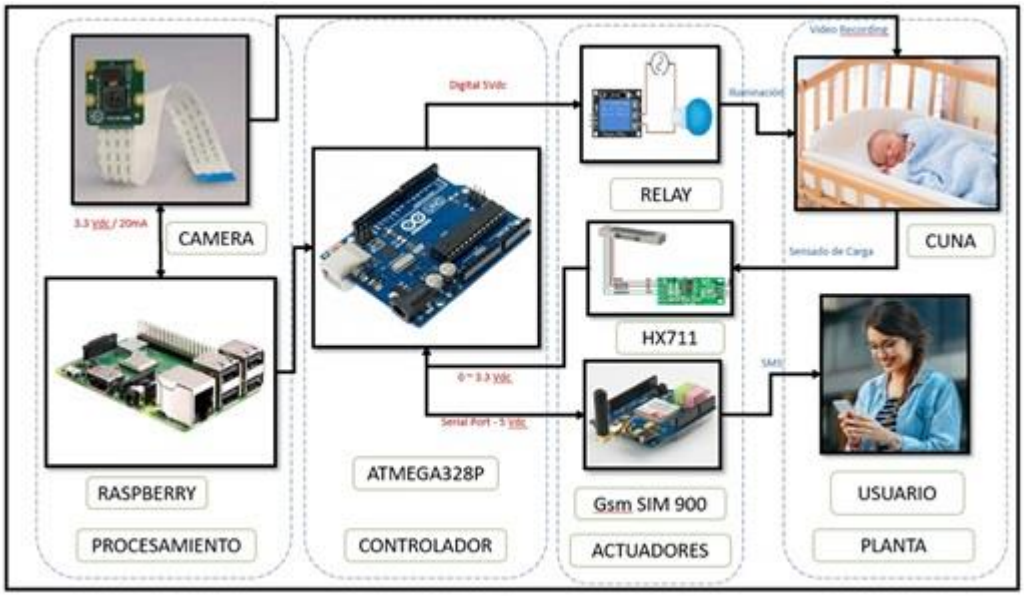


Figura 65: Diagrama de Valores de las señales electrónicas del prototipo.

Fuente: Elaboración propia

Igualmente, el prototipo mantiene un registro del peso distribuido del bebé sobre la cuna, sensando si este se posiciona en las zonas laterales. En caso de ocurrir ello, se activa la alerta de cercanía a la baranda; por lo cual existe un posible riesgo de caída si es que el bebé intenta pararse apoyándose en esta. Por ello, llegará el mensaje de texto alertando de la posición riesgosa en la que se encuentra el bebé. El ATMEGA 328P en comunicación serial con el Raspberry, aguarda la llegada del carácter para enviar dicho mensaje de texto, el cual cambia de acuerdo a la

posición detectada. Entonces, luego de enviar el mensaje de texto mediante el GSM SIM900 se activa el juguete de cuna para la distracción del bebé. Por lo cual, a partir de ese momento ya es compromiso de los responsables del bebé acudir a él para ponerlo bajo resguardo.

3.3 Funcionamiento por etapas

ETAPA 1: ATMEGA 328P – HX711: Se realiza la conexión que se muestra en la Figura 66, entre el Arduino y el HX711 para la prueba de lectura de los datos de peso.

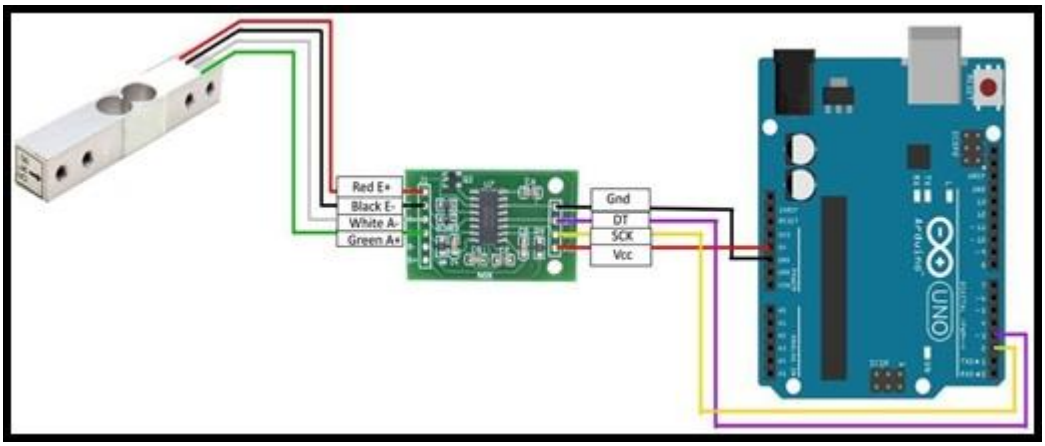


Figura 66: Diagrama de Conexiones entre Arduino y HX711

Fuente: Raúl D. (2018). Construcción de un pluviómetro digital para Arduino.

A través del ATMEGA 328p, se realizó una lectura serial del peso obtenido por el HX711, el cual se encuentra conectado al sensor de carga y cuya diferencia de potencial es la que se registra por el ADC, información enviada al ATMEGA 328P a través del HX711, tal como se muestra en la Figura 65. En los Anexos 2 y 4 se aprecian las asignaciones de pines de estos dispositivos. Por otro lado, para realizar la comparación de la cuna y la cercanía del bebé a la baranda, el parámetro de activación para las pruebas fue de 15 kilogramos; debido a que no se contó con un bebé real, la activación se realizó de forma manual; sin embargo, este peso es variable, de acuerdo al bebé monitoreado y a su edad. En la Tabla 4 se puede encontrar cómo es el cambio de talla y peso conforme los bebés van creciendo, lo cual indica que, de acuerdo a la edad, el parámetro de activación de alerta en las barandas debe ser ajustado para brindar confiabilidad de que la medición es la correcta. De acuerdo a lo mencionado en la sección 3.1.4, se requiere realizar la calibración de la celda de pesaje para obtener una medición confiable, de esta manera el proceso se realiza utilizando el programa del Anexo 5, el cual es explicado a continuación.

Básicamente, se procede a hallar el valor de la escala que se usa; es decir, se halla el factor de conversión para convertir el valor de lectura en un valor con unidades de peso. Por lo cual, es correcto destacar que la escala es diferente para cada celda y cambia de acuerdo a la forma de instalación, al peso máximo o al modelo de celda de carga; incluso, así se trate del mismo modelo de celda no necesariamente tienen el mismo valor de escala. De esta forma, primero se necesita conseguir un objeto con peso conocido, en otras palabras, se utiliza un peso patrón. Por ello, se recomienda que el peso conocido sea cercano al valor máximo del rango de trabajo de la celda de carga. De esta manera, para el caso de la presente investigación se utilizó un peso de 18Kg, debido a que la celda es de 20Kg. Por otro lado, el programa debe ejecutarse sin el peso colocado, pues al inicio del mismo se calcula la tara. Luego, después de abrir el monitor serial y esperar para que reste la tara, se procedió a colocar el objeto de 18Kg. Posteriormente, se colocó el peso sobre la balanza, y en el monitor serial se mostraron las lecturas del peso; estos valores son lecturas sin escalar, por lo que aparecen en números grandes. Inmediatamente, con el promedio de las lecturas encontradas se procedió a escalar, lo cual consistió en dividir el valor de la lectura entre el peso real. Este valor de escala, se colocó en la programación de la lectura de pesaje bajo el parámetro de "balanza.set_scale()".

A continuación, en la Tabla 4 se aprecian los pesos de acuerdo a la edad de un bebé, por lo que en la aplicación del presente prototipo se reguló el parámetro de activación de la celda de peso cada período de tiempo que el bebé incrementa su peso, con la finalidad de brindar confiabilidad a la medición y la alerta correspondiente. En la Figura 67 se puede apreciar la activación de la alerta señalada.

Tabla 4: Peso Medio y Talla de Infantes según su edad

Edad	Peso Medio	Talla
Recién nacido	3,4 kg	50,3 cm
3 meses	6,2 kg	60 cm
6 meses	8 kg	67 cm
9 meses	9,2 kg	72 cm
12 meses	10,2 kg	76 cm
15 meses	11,1 kg	79 cm
18 meses	11,8kg	82,5 cm

Fuente: Lelis G. (2019). Guía Infantil de la Salud y Embarazo

Además, en la Figura 67 se puede apreciar que se requiere un factor de calibración, el cual permite regular la lectura inicial del sensor para que inicie en 0gr; a partir de ello, se procedió con la realización de la calibración de la lectura del sensor.

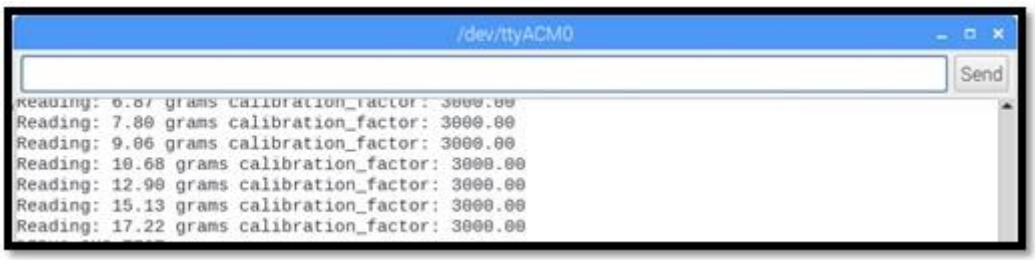


Figura 67: Datos recibidos del sensor HX711

Fuente: Elaboración propia

ETAPA 2: ATMEGA 328P – GSM SIM900: Esta etapa consiste en realizar la conexión entre el Arduino y el módulo GSM Sim900, tal como se muestra en la Figura 68 y donde se observa la prueba de operación y comunicación al envío de SMS de un número asignado por programación. Este número de teléfono es el correspondiente a los padres y/o persona responsable del cuidado del bebé, por lo cual no tiene un límite de números; sin embargo, para esta investigación solo se utilizó 1 número destino. Seguidamente, en el Anexo 3, se aprecia la asignación de pines de fábrica del módulo GSM SIM900 con el objetivo que se conozca cómo es el funcionamiento del módulo, teniendo en cuenta que los parámetros establecidos de fábrica se deben de respetar ya que al integrarse a la red móvil, la configuración no debe de ser modificada. Únicamente se utiliza la programación en el ATMEGA 328P para configurar el número destino y los pines que se utilizaron; es decir, las funciones del módulo que se requirieron, debido a que es necesario señalar que no se usan todas las propiedades aplicativas del mismo, sino únicamente el envío de SMS.

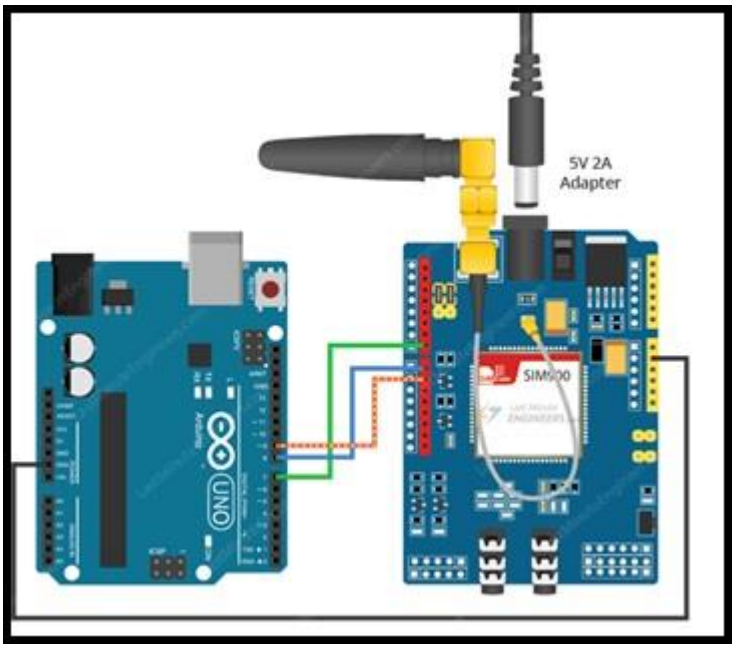


Figura 68: Diagrama de conexiones entre ATMEGA 328P y el GSM SIM900

Fuente: Elaboración propia

En la Figura 69 se puede visualizar el módulo GSM SIM900 energizado y aguardando la conectividad con la red telefónica del chip GSM que tiene en su interior.

Figura 69: Encendido del módulo GSM SIM900 y operatividad

Fuente: Elaboración propia

Asimismo, el módulo GSM tarda 10 segundos en registrarse con la red para iniciar el canal de comunicaciones de envío de mensajes de texto. Una vez registrado en la red, el led parpadea a menor frecuencia, lo cual indica su registro y operatividad. Con ello, el prototipo se encuentra a la espera del carácter enviado mediante la comunicación serial entre el Raspberry y el ATMEGA 328P, indicándose así la posición en la que se encuentra el bebé para poder determinar el contenido del mensaje que será enviado.

3.4 Prototipo integrado

El prototipo completo integra todos los dispositivos, por lo cual trabaja y aguarda el resultado del procesamiento de imagen por parte del Raspberry Pi. De ello también depende la activación del relé, el cual se activa de acuerdo con el peso sensado por la celda de pesaje y "traducido" el valor por el HX711. A su vez, el Raspberry y el procesamiento de imagen dan el inicio al funcionamiento del prototipo. A continuación, se observa una fotografía del prototipo en la Figura 70, la cual consta de la cuna, las celdas de peso, el Raspberry, el ATMEGA 328P, el juguete distractor de cuna, el módulo GSM SIM900 y el soporte donde se ubica la cámara. Cabe señalar que el juguete de cuna escogido posee un foco giratorio y sonido, y es establecido por los usuarios finales. Para la presente investigación, el diseño como producto final y la ubicación real de cada componente en la cuna no se realiza, debido a que el objetivo es buscar la funcionabilidad del prototipo.

Figura 70: Integración del Prototipo

Fuente: Elaboración propia

En la Figura 71 se aprecia únicamente la etapa de control del prototipo. En la posición que se encuentra la celda de pesaje se realizaron las pruebas de su funcionamiento, ya que como se ha mencionado, la lectura del peso fue arbitraria y generada de forma manual.

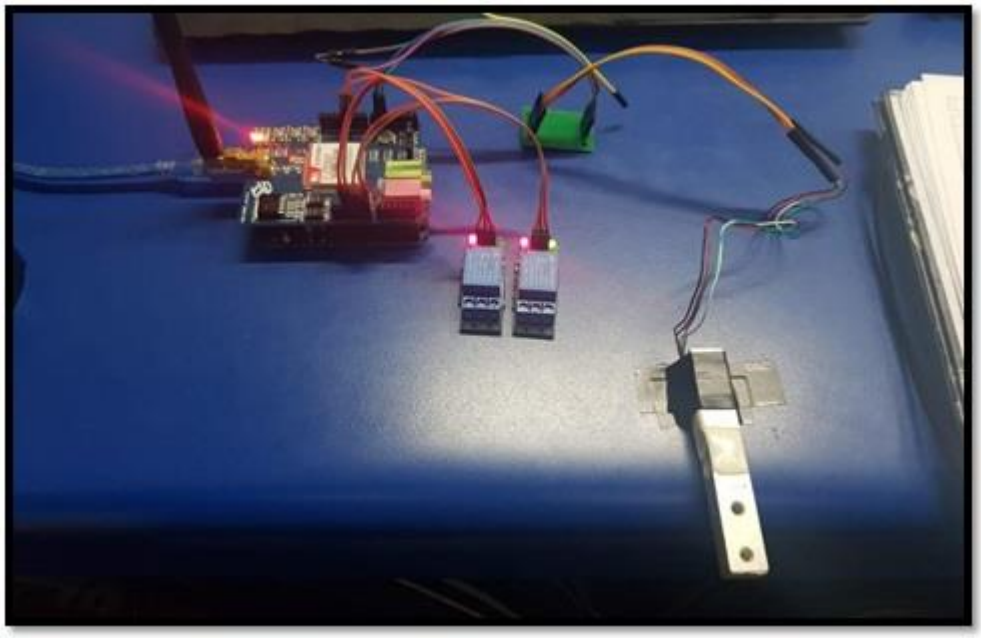


Figura 71: Funcionamiento de la etapa de control del Prototipo

Fuente: Elaboración propia

Asimismo, para poder asegurar la correcta comunicación entre ambas tarjetas electrónicas (Raspberry Pi y ATMEGA 328P), fue necesario realizar pruebas de envío de caracteres, ya que el programa final envía letras que están asignadas a realizar una acción específica para determinar cómo ha de actuar el prototipo.

3.5 Funcionamiento del Raspberry Pi y el PiNoir Camera V2

Para el funcionamiento, el Raspberry Pi debe de tener habilitada la cámara periférica PiNoir. Y esto se logra en la configuración de dispositivos del mismo Raspberry. Luego de ello, en la programación se debe colocar las librerías que permiten el funcionamiento correcto de la cámara "import picamera" e "import time". La segunda librería corresponde a la toma de intervalos de tiempo o pausas para la captura de las imágenes con el comando "time.sleep( )". Además, el sostenimiento y el enfoque de la cámara deben de ser adecuados para que se mantenga en una posición fija. De esta manera, para la toma de imágenes de la base de datos de muestra, se utilizó un script que permitió la adquisición y almacenamiento de imágenes en la memoria del Raspberry Pi. En la Figura 72 se muestra la posición de la cámara en la que se fueron capturadas todas las imágenes.

Figura 72: Posicionamiento y Soporte de la Cámara

Fuente: Elaboración propia

Asimismo, este soporte necesita ser adaptado estructuralmente para que la colocación del Raspberry y ATMEGA 328P sea adecuada. Queda claro, que en la presente investigación no se encuentra contemplado la realización del soporte final.

3.6 Información de la Muestra

En la presente investigación se usó como base de pruebas un modelo de cuna de madera color blanco, con una colcha de franjas bicolores naranja y rosa. Las medidas que posee la cuna son de 138x110x80cm tal como lo muestra la Figura 73. Asimismo, el muñeco que se utilizó tiene medidas de 20x40cm, las cuales se aprecian en la Figura 74.

Figura 73: Medidas de la Cuna de Muestra

Fuente: Elaboración propia

Figura 74: Medidas del Muñeco de Muestra

Fuente: Elaboración propia

Cabe mencionar que no se realizó ninguna prueba con un infante real, debido a que es difícil conseguir el permiso de los padres para la realización de tales pruebas de estudio. Además, la realización de pruebas con bebés reales no es parte del objetivo de la presente investigación.

3.7 Análisis de Rentabilidad Económica del Prototipo

Se presenta en la Tabla 5 el presupuesto utilizado para el desarrollo del prototipo.

Tabla 5: Presupuesto del Prototipo

Material	Costo (S/.)
Raspberry Pi Model B+	250.00
Cámara Pi NOIR	100.00
ATMEGA 328P	50.00
HX711	10.00
Celda de Pesaje 20Kg.	20.00
Módulo GSM SIM900	200.00
Chip de telefonía móvil (más recarga)	20.00
Juguete de Cuna	20.00
Relé	10.00
Total (S/.)	**680.00**

Fuente: Elaboración propia

En la Tabla 6, se aprecian distintos equipos que ofrecen servicios similares respecto al monitoreo de un bebé, incluyéndose sus características y costo. Asimismo, se destaca que estos productos son vendidos en el mercado peruano.

Tabla 6: Características y Costo de Equipos Similares en el Mercado Peruano

Nombre de Producto	Características	Costo (S/.)
Cámara Para Bebé Monitoreo Wifi Internet	Ver imágenes en HD, hablar desde el celular a través de la cámara, girar la cámara desde el celular, memoria SD para grabar imágenes.	300.00
Dormi Baby Monitor (solo APP)	Control remoto de dispositivos, grabación en Tiempo-Real, filtros de audio inteligentes, múltiples dispositivos soportados, monitoreo ilimitado con el plan Premium.	100.00
Monitor De Video Para Bebé 3.5 Con Wi-fi Connect – Motorola	Tecnología inalámbrica 2.4GHz FHSS (para visualización local) y conectividad inalámbrica Wi-Fi (para una visualización remota), Conexión Wi-Fi con zoom digital, Visualización a color de 3.5", Visualización de temperatura del ambiente, Visión nocturna por infrarrojos, Unidad de padres con dos vías de comunicación y out-of-range y alerta de batería baja, Cuenta con micrófono de alta sensibilidad con LED Indicadores de nivel de sonido, Rango hasta de 300 Metros, Notificaciones de movimiento, y la temperatura.	700.00 (mensuales)
Ibaby Cámara De Video Digital Inalámbrica Para Bebé	Máxima área de cobertura con giro de 360 grados e inclinación de 110 grados, F4.2 milimétricas y F2.0 Iris fijo de lentes (2 megapixels), jugar	650.00

	con música relajante para el bebé usando el reproductor de música de iBaby y grabar su voz en OFF de su historia favorita de acostarse con grabación de voz de iBaby, vigilar al bebé en HD claro día o la noche de en cualquier lugar y capturar imágenes de una calidad cristalina. Sensores CMOS.	

Fuente: Elaboración propia

Considerando que el precio del prototipo, incluyendo las ganancias, fue de S/.750, entonces al compararse con los productos actuales resultaría con un precio elevado; sin embargo, se aprecia que el prototipo planteado ofrece el sistema de alerta a los padres, mientras que en los otros productos se debe estar pendiente de los movimientos del bebé. Por otro lado, el prototipo planteado está diseñado para determinar el momento en que el bebé se encuentra en una posición de peligro cuando se encuentra dormido, o bien cuando se encuentra cerca a las barandas de la cuna. Por ello, ninguno de los productos de la muestra anterior ofrece esa característica, solo llegan a ofrecer alertas por los movimientos del bebé.

Bajo las condiciones anteriores, se demuestra que el presente prototipo, si bien posee un precio elevado a los productos existentes en el mercado, ofrece beneficios que otorgan mayor seguridad al bebé cuando se encuentra solo en su habitación. Además, se agrega que los padres y/o responsable del cuidado del bebé pueden realizar otras actividades y no tienen que encontrarse pendientes del estado del bebé. Por lo que la factibilidad del presente prototipo, en el caso que se desee comercializar, es aceptable debido a que es una solución innovadora.

CAPÍTULO IV: PRUEBAS Y RESULTADOS

4.1 Prueba y Resultado de envío de Caracteres

La Figura 75 muestra la programación para el envío de caracteres que se utilizó en el Raspberry Pi (derecha) y en el ATMEGA 328P (izquierda). El comando que habilitó la comunicación serial fue "Serial.begin()", donde 9600 es la velocidad de datos en bits por segundo (baudios). Para comprobar que la comunicación fue establecida y no se perdieron los datos, se utilizó "Serial.available", donde el ">0" significa que los dispositivos (Raspberry y ATMEGA 328P) se encuentran conectados. Para el caso del Raspberry, se necesitó importar la librería "serial"; luego de ello, fue necesario declarar con quién se establece la comunicación. Por eso, se contó con la variable "ard" que representó a "serial.Serial()" y por lo cual necesitó de 2 parámetros. El primero fue el puerto de conexión en el que se encontró, y el segundo la velocidad en baudios; para esta prueba se aprecia que el conector serial fue reconocido en "/dev/ttyACM2" con la velocidad de 9600 baudios. Luego de ello, con el comando "ard.write()" se envió el carácter que se deseó.

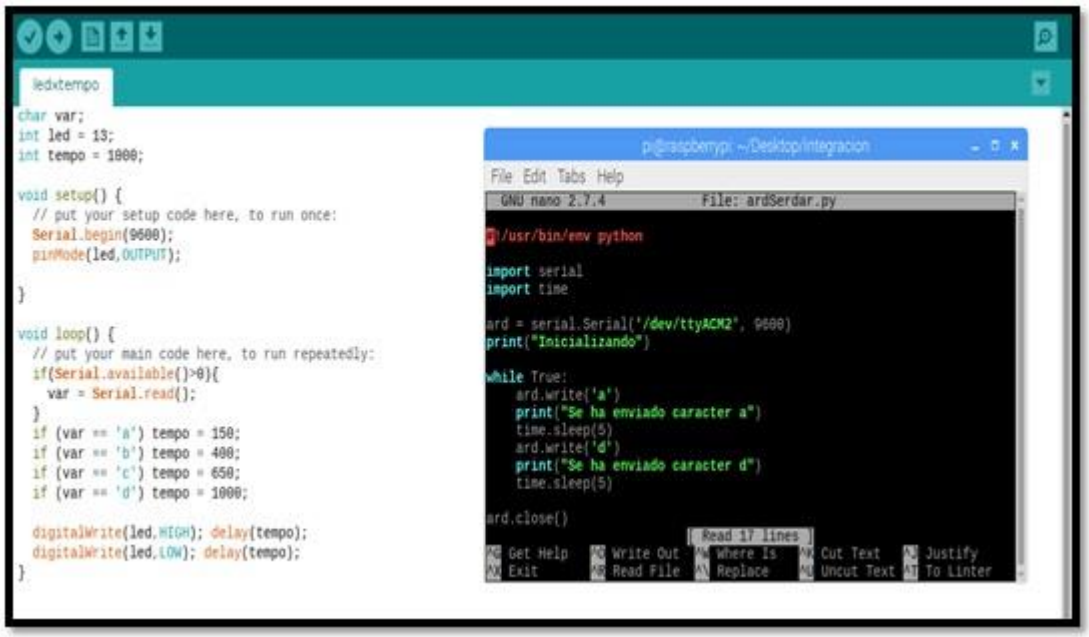


Figura 75: Envío de Caracteres entre Raspberry y ATMEGA 328P

Fuente: Elaboración propia

En la Figura 76 se muestra el resultado del envío de los caracteres "a", "b", "c" y "d"; realizándose de manera satisfactoria bajo la programación ya explicada anteriormente.

Figura 76: Confirmación de envío de Caracteres

Fuente: Elaboración propia

4.2 Prueba y Resultado de Funcionamiento de Celda de Pesaje

Respecto a las pruebas realizadas con la celda de pesaje, la Figura 77 muestra la respuesta de la misma cuando sensa 15.13Kg, por lo cual se procede a realizar el envío del mensaje de texto correspondiente.

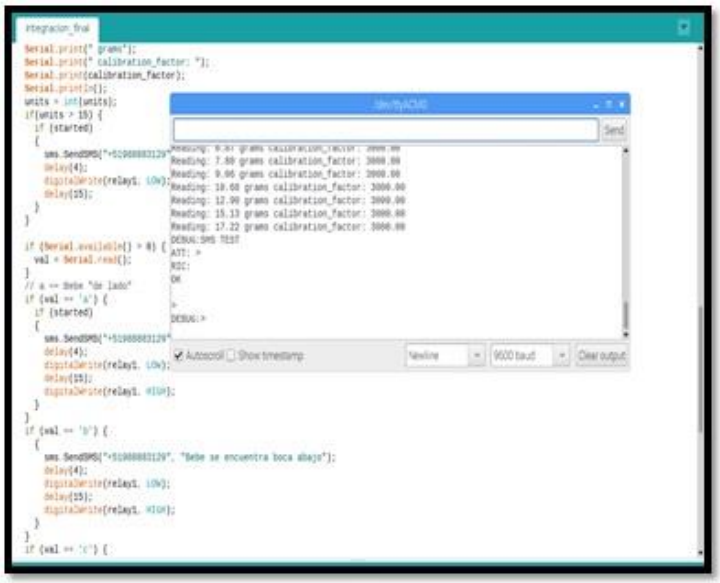


Figura 77: Detección de 15Kg por la Celda de pesaje

Fuente: Elaboración propia

4.3 Pruebas y Resultados con el Prototipo Integrado

Las figuras 78 y 80 corresponde al resultado coincidente al aplicar la TM_SQDIFF_NORMED que se obtiene de la detección de la posición número 1

y 4 respectivamente; mientras que en la Figura 79 se aprecia el resultado numérico del procesamiento de imagen de dicha posición.

Figura 78: Resultado del procesamiento de imagen al detectarse posición 1

Fuente: Elaboración propia

Figura 79: Resultado Numérico del procesamiento de imagen al detectarse posición 1

Fuente: Elaboración propia

Asimismo, la Figura 81 corresponde al resultado numérico de la detección de la posición número 4.

Figura 80: Resultado del procesamiento de imagen al detectarse la posición 4

Fuente: Elaboración propia

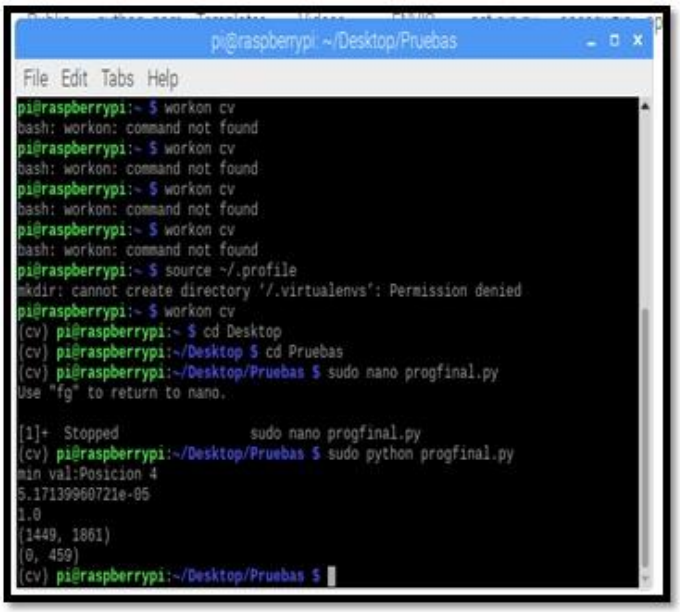


Figura 81: Resultado al detectarse posición 4

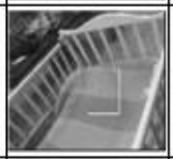
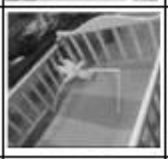
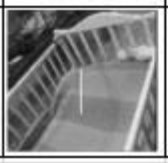
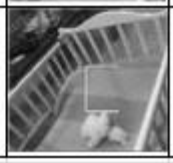
Fuente: Elaboración propia

Para poder llevar una contrastación del funcionamiento correcto del prototipo, fue necesario realizar pruebas en posiciones que no son iguales a la base de datos Template. Es por ello que se procedió a colocar al bebé en diversas posiciones para identificar la robustez y margen de error del prototipo, las cuales se presentan a continuación.

4.4 Pruebas y Resultado con posiciones Distintas

En la Tabla 7 se observan las pruebas realizadas con sus resultados obtenidos.

Tabla 7: Resultados de pruebas de posiciones distintas

ARCHIVO	POSICIÓN	DETECCIÓN	ARCHIVO	POSICIÓN	DETECCIÓN
Abajo 1		SI	Caído 3		NO
Abajo 2		SI	Caído 4		NO
Abajo 3		SI	De lado 1		SI
Abajo 4		SI	De lado 2		SI
Arriba 1		SI	De lado 3		SI
Arriba 2		SI	De lado 4		SI
Arriba 3		SI	Aleatorio 1		NO
Arriba 4		SI	Aleatorio 2		SI
Caído 1		NO	Aleatorio 3		NO
Caído 2		NO	Aleatorio 4		NO

Fuente: Elaboración Propia

Se puede apreciar que, de las 20 muestras, 7 tienen un resultado equivocado; es decir el 35% del total. Sin embargo, los errores encontrados se deben a que la posición en la que se encontraba el infante al momento de realizar la captura con la cámara, no se encontraba dentro de las imágenes Template, lo cual ocasionó que el resultado se encuentre basado en el color del fondo de la imagen; por lo tanto, la frazada y el color de la ropa también influye en el resultado otorgado por el algoritmo. Entonces, para resolver estos errores se utilizó una base de datos fotográfica adecuada para poder captar todos los movimientos del bebé en su cuna. En la Figura 82 se tiene una estadística con los resultados correctos e incorrectos obtenidos. El "NO" significa que no detectó correctamente la posición del bebé, mientras que el "SÍ" lo hizo.

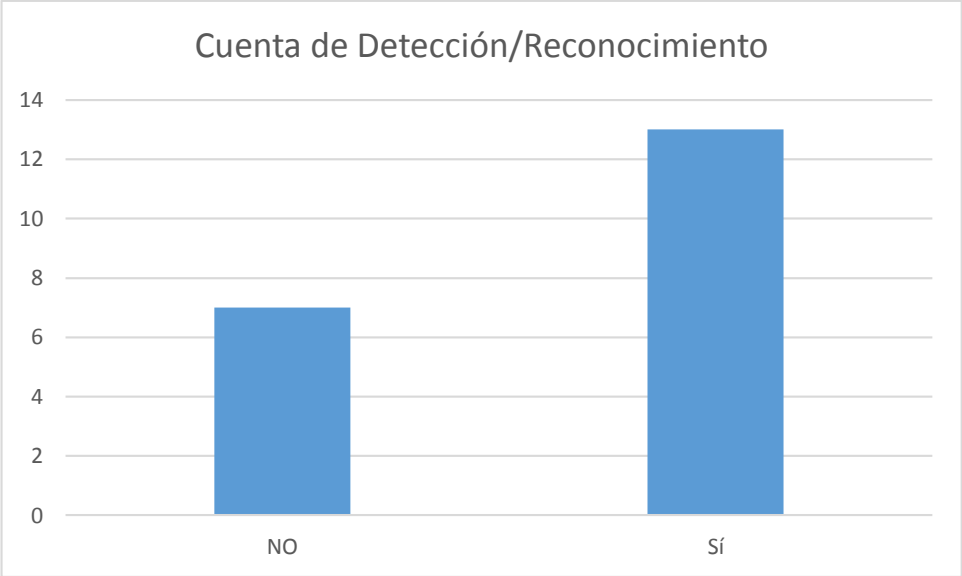


Figura 82: Resultados de la Detección de Posición del bebé

Fuente: Elaboración propia

Luego, se procedió a analizar los datos erróneos de las pruebas anteriores. En la Figura 82 se aprecia la cantidad de Buenos y Malos resultados. Por lo que, en la Figura 83, se utilizó la data obtenida de los resultados fallidos para conocer qué resultado es el que se obtuvo. En el eje X se tienen las 7 pruebas fallidas, mientras que en el eje Y las 4 posiciones distintas establecidas.

Figura 83: Resultado del Número de Posición

Fuente: Elaboración propia

Además, se aprecia que todos los resultados que no reconocen la posición correcta del bebé en su cuna detectan la posición 1. Como se ha explicado anteriormente, ello se debe porque las imágenes de muestra entre las 4 posiciones iniciales establecidas, la posición 1 es la que posee más parecido en cuanto al color y fondo de las capturas que se han tomado como muestra.

CONCLUSIONES

1. Se diseñó y se programó el prototipo de monitoreo diurno en tiempo real utilizando visión artificial, con la finalidad de reconocer la posición de un bebé mientras se encuentra en su cuna. De esta manera, con el uso de los componentes descritos en la sección 3.4, se pudo apreciar que el prototipo no ha sufrido modificaciones durante el desarrollo del proyecto. Pues, debido a que conforme se desarrollaron las etapas del funcionamiento y programación, el mayor reto fue lograr la comunicación entre el Raspberry y el ATMEGA 328P. Así como también, se pudo lograr que el prototipo actúe autónoma y correctamente, y se comprobó el correcto funcionamiento del monitoreo de bebés en su cuna. En cuanto al porcentaje de error del prototipo fue del 35%, teniendo en cuenta que las muestras son posiciones no establecidas como base de datos; asimismo, brinda robustez y confiabilidad al prototipo.

2. Se programó el algoritmo de procesamiento de imagen basado en la técnica Template Matching teniendo como función la "Diferencia Normalizada" para que se pueda reconocer la posición del bebé. El desarrollo fue realizado utilizando el lenguaje de programación Python y su librería OpenCV. Así como también, se utilizó una comunicación serial y la cámara Pi NOIR. Para el caso de la programación del ATMEGA 328P se programó en lenguaje C+ para que reciba la data enviada por el Raspberry y de acuerdo a ella, se activen los actuadores (GSM Sim900 y juguete distractor de cuna).

3. Se logró desarrollar el envío de alertas mediante mensajes de texto y utilizando el Shield GSM Sim900, así como la activación de un juguete ante un posible peligro del bebé mientras se encuentra en una cuna de tamaño estándar. Todo ello fue abordado en la sección 3.2 correspondiente a la descripción del prototipo, así como en la sección 3.4 correspondiente a Prototipo Integrado.

RECOMENDACIONES

1. Se recomienda que el sensor de pesaje se someta a una calibración previa a su uso, para establecer un adecuado factor de pesaje y brindar la información confiable.

2. Se recomienda que periódicamente se recargue con saldo el chip GSM, pues como se envía un mensaje de texto continuamente se estaría consumiendo el saldo del mismo. En el caso que no existiera saldo, el funcionamiento del prototipo no se ve afectado; sin embargo, el mensaje de texto alertando a los padres y/o responsable no sería posible enviarlo.

3. Se recomienda realizar un estudio previo del bebé durmiendo y fotografiarlo con el prototipo, para tener la base de datos del Template Matching.

4. Se recomienda escoger 2 números telefónicos para el envío de alertas, puesto que a pesar de ser establecidos en la programación del módulo GSM SIM900, no pueden ser cambiados por el usuario. Además, que ante una mayor cantidad de números telefónicos destinos, se estaría consumiendo más rápido el saldo del chip.

5. La recomendación para el procesamiento de imágenes es sobre el color de fondo y la ropa del bebé, ya que influyen en el reconocimiento de la posición, hay que tener en cuenta los cambios bruscos de colores al momento de realizar una muda de ropa tanto a la cuna como al bebé.

REFERENCIAS BIBLIOGRÁFICAS

Atmel, A. (2015). 8-bit Microcontroller with 4/8/16/32K Bytes. *System Programmable Flash*, *13*(1), pp. 1.

Avia, C. (2014). *24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales*. Madrid, España: Editorial Amat.

Brangui, S. (2015) An ergonomic comprehensive system for monitoring and control of baby environments. *6th International Conference on Information and Communication Systems (ICICS)*, *6*(2), pp. 251–256.

Collazos, C. (2010). *Procesamiento de imágenes.* México: Editorial Almaida.

Crespo, R. (2012). *Opencv: Tratamiento de imágenes.* Argentina: Cervantes.

Diaz, J. (2016). Celdas de peso con Arduino. *Mi Arduino*, *1*(1), pp. 11–19.

Dumas, A. (2017). La posición adecuada para dormir a tu bebé. *Baby World*, *4*(2), pp. 5–20.

Durand, G. (2010). Sim 900. *Smart Machines, Smart Decision*, *3*(2), pp. 34–47.

Elmas, F. (2017). Rocking motion of the baby sleeping on the mother's lap: Modeling and prototype automatic swing cradle design. *Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, *5*(1)pp. 1–5.

Escolano, F. (2003).Inteligencia artificial. Modelos, técnicas y Áreas de aplicación. *Inteligencia Artificial*, *2*(1) pp.30–50.

Esqueda, D. (2002). *Técnicas de vision Artificial.* Buenos Aires, Argentina.

Farnell, G (2016). *Raspberry PI 3 Model B*. México: Contribuciones Mex.

García, J. (2011). Arduino+matlab: adquisición de datos. *Microcontroladores*, *3*(6), pp. 48–71.

Ginemo, R. (2010). *Estudio de técnicas de reconocimiento facial.* Sao Paulo, Brazil.

Halfacree, G. (2015).*Diseñar y desarrollar proyectos avanzados de visión por computadora usando OpenCV con Python.* Valparaiso, Chile.

Mocq, F. (2016). *Raspberry pi 2: Utilice todo el potencial de su Nano-ordenador*. Río de Janeiro, Brazil.

Mordvintsev A. (2013).*Template Matching: Theorie.* Toronto, Canadá.

Lelis, G. (2019). *Pesos y estaturas del bebé, niño y niña.* Madrid, España.

Ou, W. (2013). Video based vomit and facial foreign object detections for baby watch and safety. *1st International Conference on Orange Technologies (ICOT), 1*(2), pp. 13–20.

Raúl, D. (2018). *Construction of a digital rain gauge for Arduino.* Montreal, Canadá.

Rojas, T. (2008). Sistema de visión por computadora para la detección de objetos esféricos a través de la transformada de Hough. *Revista Ingeniería UC, 3*(1), pp. 77–87.

SIMCOM (2015). The GSM/GPRS Module for M2M applications. *GPRS applications, 4*(1), pp. 2.

Stoitchkov, D. (2017).Analysis of Methods for Automated Symbol Recognition. *Technical Drawings. 3*(1), pp. 4–5.

Tinyos, E. (2017). 2 Channel 5V Relay Shield module. *Electronics, 2*(3), pp. 88–92.

Uit, U. (2000). Recomendación UIT-r m.1073-1. *Recomendaciones UIT*, *3*(1), pp. 17-33.

Uit, U. (2003). Recomendación UIT-r m.1036-2. 21. 1. *Recomendaciones UIT, 4*(1), pp. 30–50.

UNICEF. (2012). Guía para la familia sobre el embarazo y la Primera infancia. *Crecer, 3*(1), pp.16–19.

UNICEF. (2013). Clave para la primera infancia de 0 a 3 años. *Desarrollo emocional*,*4*(1), pp. 24–26.

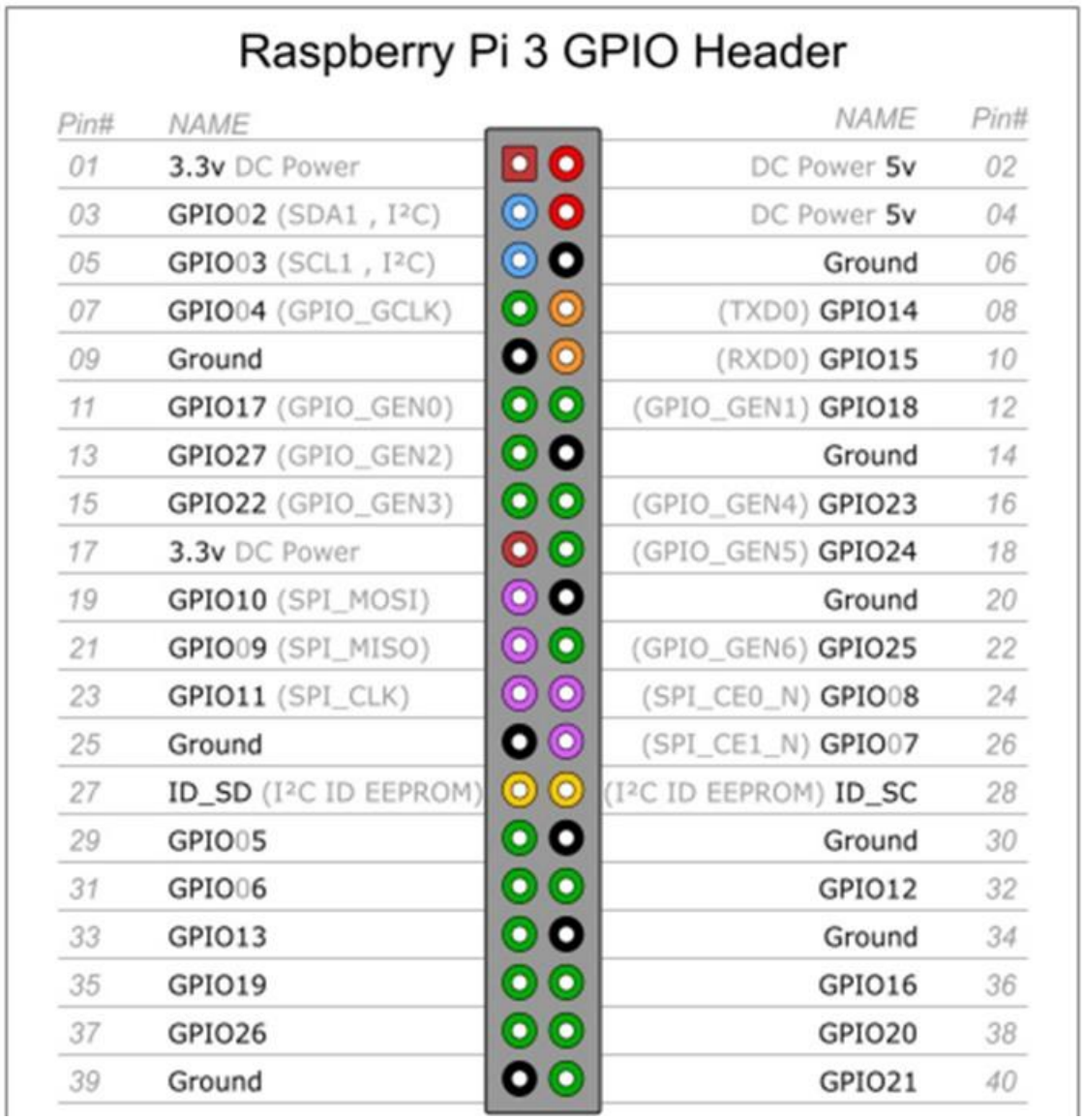
Upton, E. (2013). Raspberry pi 200 ejercicios prácticos. *In Inicios de Raspberry, 2*(1), pp. 49–52.

Valle, A. (2015). Tutorial: Shape detection. *Technology Robotix Society, 2*(2), pp. 19–26.

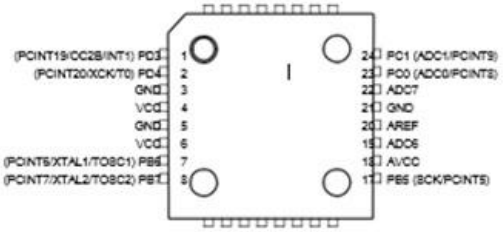
Zegarra, C. (2011). Tutorial: Conectando un módulo relé con Arduino. *Arduino, 1*(1), pp. 32–38.

ANEXOS

Anexo 1.- Configuración de Pines del Raspberry Pi 3 modelo B+

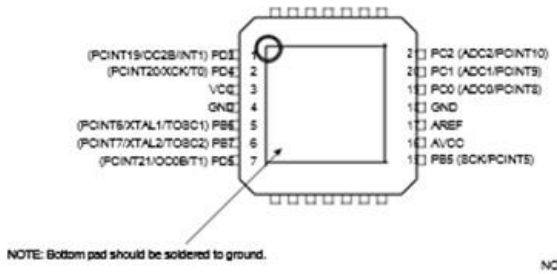


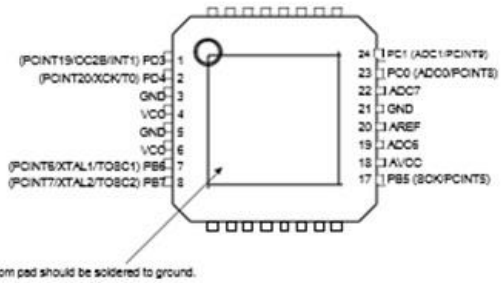
Anexo 2.- Configuración de pines de ATMEGA 328P



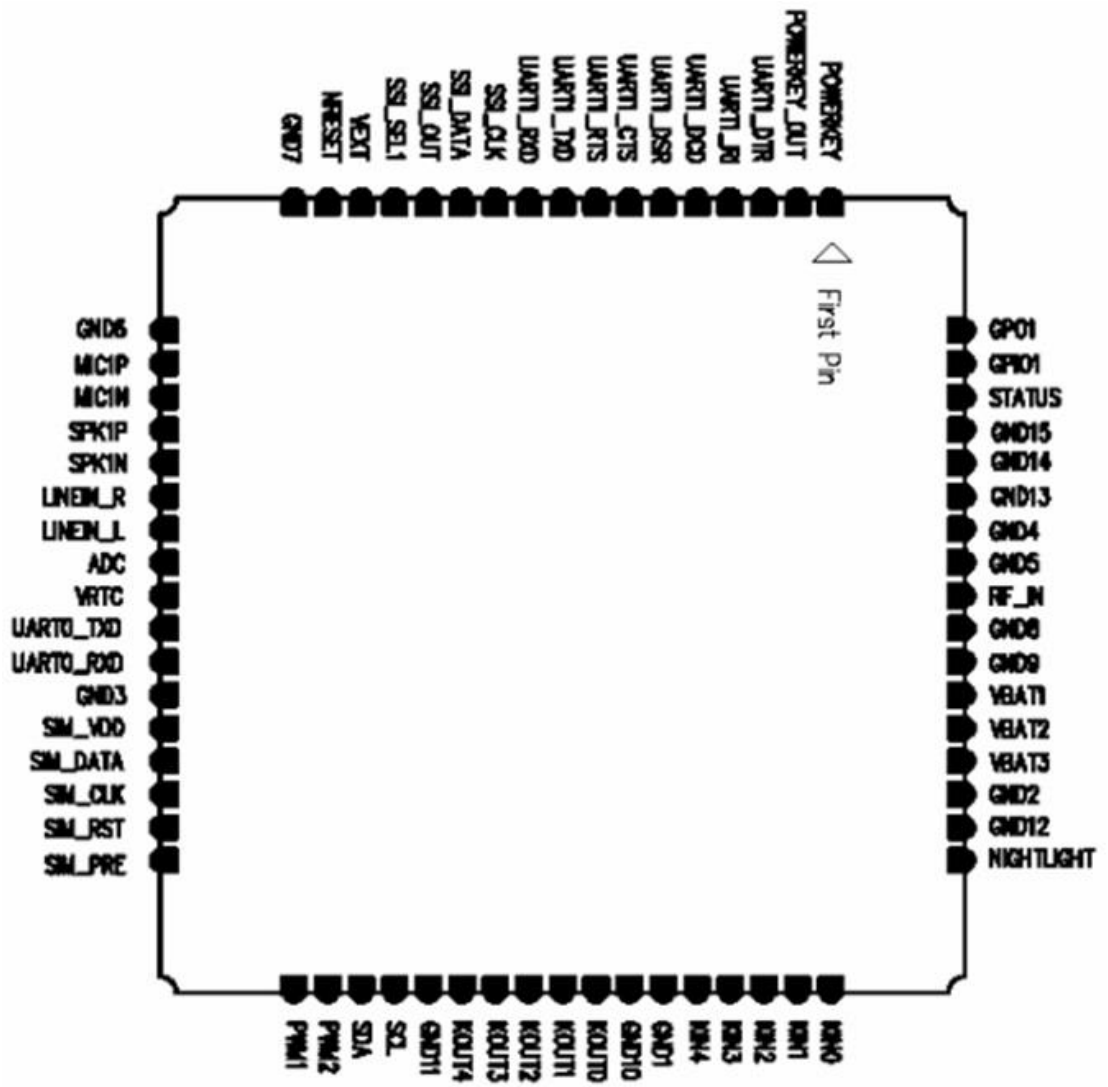
28 MLF Top View

32 MLF Top View

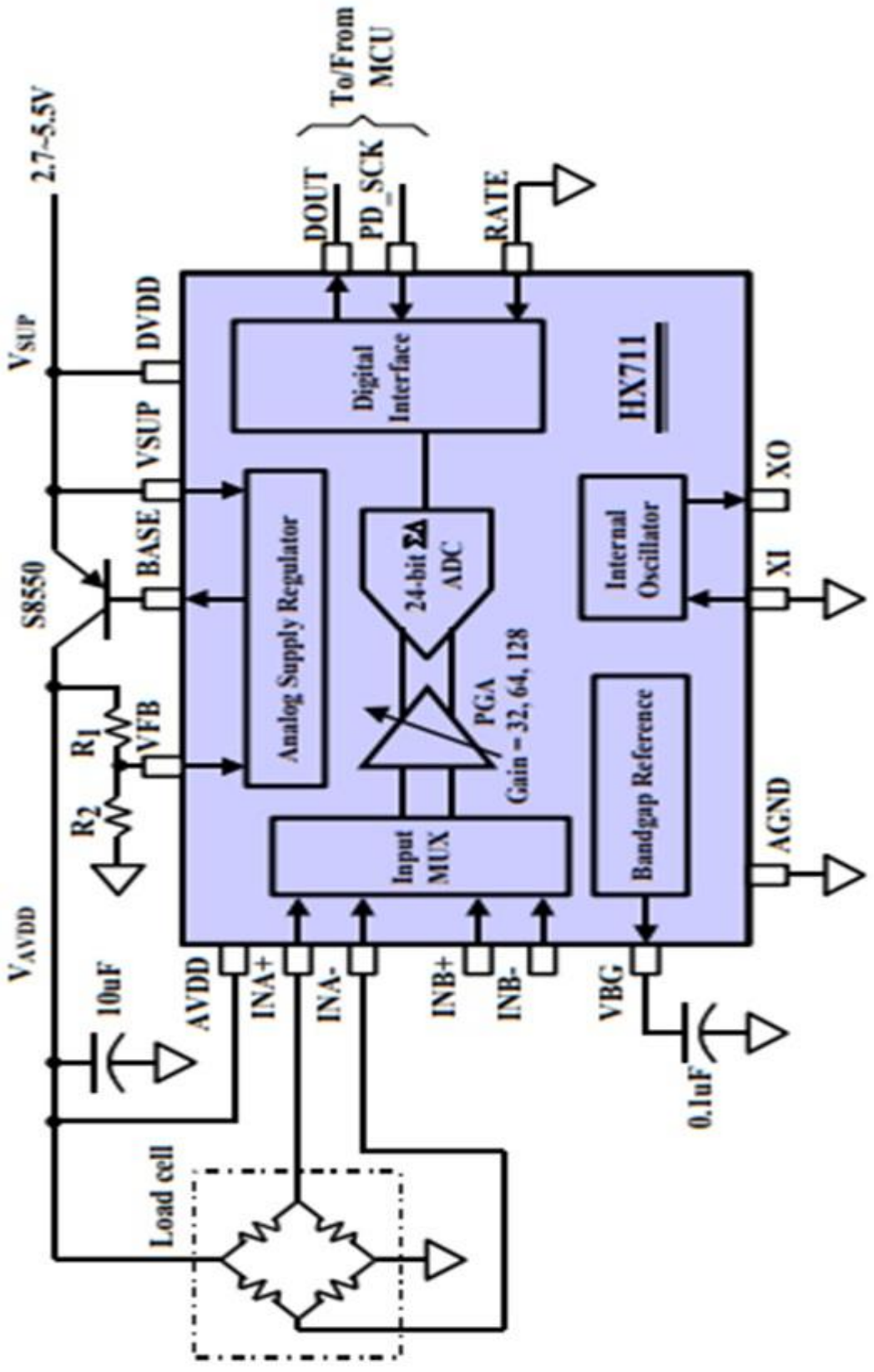




Anexo 3.- Configuración de pines de Módulo GSM SIM900



Anexo 4.- Configuración de Transductor Hx711



Anexo 5.- Programa para la Calibración de la Celda de Pesaje

```
#include "HX711.h"
#define DOUT  A1
#define CLK  A0
HX711 balanza(DOUT, CLK);
void setup() {
  Serial.begin(9600);
  Serial.print("Lectura del valor del ADC:  ");
  Serial.println(balanza.read());
  Serial.println("No ponga ningun  objeto sobre la balanza");
  Serial.println("Destarando...");
  balanza.set_scale(); //La escala por defecto es 1
  balanza.tare(20);       //El peso actual es considerado Tara.
  Serial.println("Coloque un peso conocido:");
  }
void loop() {
  Serial.print("Valor de lectura:  ");
  Serial.println(balanza.get_value(10),0);
  delay(100);
}
```