IoT-Enabled Cattle Health and Location Monitoring System

THISURA RAJAPAKSE
Department of Computer Engineering,
General Sir John Kotelawala Defence University,
SRI LANKA.

MWP MADURANGA
Department of Computer Engineering,
General Sir John Kotelawala Defence University,
SRI LANKA.

MB DISSANAYAKE
Department of Electrical & Electronic Engineering,
Faculty of Engineering,
University of Peradeniya,
SRI LANKA.

Abstraction **-** In human-animal interactions, the capabilities of the IoT concept become a promising game changer. The owners of animal farms can already utilize smart sensors to identify ways to keep an eye on the health, whereabouts, behavior, and/or environment of their animals. On the other hand, despite this notion being in use for years, there are still some concerns that need to be resolved. In this work, we contribute to the design of a state-of-art wearable collar for cattle that can monitor the location and health conditions remotely in large-scale farms. The proposed collar is designed on a microcontroller with GSM/GPRS communication modules. Therefore, the wearable collar can communicate directly with the IoT server via 3G/4G or 5G mobile base stations and can be used in robust environments. Further, power-saving techniques have been investigated to prolong the lifetime of the wearable collar. Evaluation of the performances of the devices has been presented in this paper.

Keywords: - Internet of Things (IoT), Animal Health Monitoring, 5G, Smart Farming, Sensors

Received: April 9, 2022. Revised: October 29, 2022. Accepted: November 24, 2022. Published: December 31, 2022.

1. Introduction

Advances in Technology have altered the agriculture sector to improve the service rendered to the customers as well as the suppliers. Most processors in this domain have become more accessible, and time- and labor-efficient with the use of sensors, equipment, gadgets, and information technology. By implementing technology-ready devices, farmers create new avenues and operate in a productive and efficient atmosphere. Animal husbandry will be more convenient and manageable thanks to new technological instruments and techniques. Additionally, daily management decisions at the farm and animals can be efficiently configured using modern technology tools. The profitability and quality of the final output will be directly impacted by humans.

Animal and milk production volumes are influenced by a variety of variables, including the animal's genetic makeup, environment, exposure to diseases, nutrition, climate, age, and season. The stability of the economy and the safety of the nation's food supply will be maintained by keeping an eye on animal health and taking the required action when necessary. Due to trade restrictions, animal slaughter, and subsequent disease eradication measures, animal disease outbreaks can have a direct impact on the nation's economy. Additionally, this may have an impact on world trade, agricultural sector stability, and public health.

Animal health is typically monitored manually on farms. The primary factor reducing daily production in farms is the manual health monitoring system. The disadvantages of a manual system are numerous. There is a paucity of expertise regarding early disease impact detection in animals. For instance, a late diagnosis of a disease will result in expensive medical treatment. In large farms, manual animal data management is a time-consuming, labor-intensive task. Additionally, in the manual system, the decision-making process is frequently erroneous and imprecise, and it is highly subjective. Some systems are highly expensive, and some farmers find it difficult to afford such systems. Furthermore, most of these tech-based systems are designed for large farms and it is practically challenging to adopt them in small farms. Therefore, this research mainly focused on creating a monitoring system that is accurate and economical.

This research proposes an IoT-based Animal Health Monitoring System, to address above mentioned issues encountered when raising livestock. This system was designed and created as individual nodes, making it inexpensive for smallscale farmers to buy the exact quantity needed. The farmer can manually install the device and attach it to the monitoring web portal fairly easily by following the instructions.

The idea of a base station is entirely dropped in this design. Using GSM/GPRS technology, Node can interface directly with IoT servers. Direct communication means that there are no restrictions on distance. If the system notices an abnormal animal health condition, it will automatically send an alarm to the farmer through email notification. Because of this particular feature, the system is more suitable for small farms where the individual animal monitor is practical.

Also, in our proposed design, contrary to other designs in the literature, we measure the body temperature and heart rate of the animal. These two measures alone, help to diagnose the presence of infection in animals. Early diagnosis of potential infections would assist the farmers in better managing their herd and stopping the spreading of the disease through isolation of the infected.

Also, a gas sensor is included in this system to identify the air quality around the cattle. Maintaining good oxygen levels in the cowshed will help to increase productivity, while poor air quality indication would again assist the farmers to take preventive measures and improve the management of their livestock. Furthermore, this system is energy-efficient, cost-effective, and easy to use.

The paper is structured as follows, Section 1 gives an introduction to the research problem, and section 2 presents the literature review highlighting the related works. It is followed by the design and implementation of experimental results, and future works. The conclusion section concludes the paper.

2. Related Works

Different challenges are addressed by the systems that are now in use. Different components, characteristics, and technologies were used by these systems. The below summarizes specific IoT systems from the literature about the issue examined in this literature.

Vannieuwenborg et al. contributed to Designing and evaluating a smart cow monitoring system from a techno-economic perspective, This study is about developing an IoT-based precision dairy monitoring technology to overcome various factors that affect overall economic performance in the dairy industry such as reproductive health problems, diseases, and low detection rate of insemination movement. This system consists of several components and applications. Smart collars include a temperature sensor, Ultra-Wide Band base location tag, acceleration sensor, and barometer. Using a magnetic induction collar is capable of wireless charging. For data communication between the collar and base station, this system used LoraWAN protocol and technology. The base station will upload the communicated data to the cloud. Because of LoraWAN technology collar can do instance over-the-air firmware updates. Smart ear tags use to collect the temperature data and communicate with the collar via MI. NFC function is included in this tag for setup, paring, and scanning processes. Charging points are installed on drinking points feeding points and milking points to charge supercapacitors in the collar. Management platforms will help farmers to keep tracking their cows. If any threshold value is exceeded, the farmer will get notified. Mobile applications will help to locate specific cows on the farm. Using NFC farmers can access cow journals and data history. The cloud will work as the main repository and analyze all data. This allows system management via various types of interfaces (management platform and mobile application)[1].

Faruq et.al. carried out a study for a health management system for dairy cows, that is capable of detecting and handling diseased cows. This system consists of a monitoring and detecting system. For application utilization purpose system combine the Internet of Things (IoT) with Intelligent System technology. The main purpose of the monitoring system is to collect data from the sensors at each node. Each node consists of Arduino Mini Pro, ESP32, MAX30100 Heartrate sensor, and MLX90614 Temperature sensor. Raspberry PI is used in this system as a gateway. All the nodes send data to the gateway before sending it to the server (MQTT protocol). Raspberry PI will send the average value of each data and each node to the server every 30-second time gap (HTTP protocol). An intelligent system is a combination of collecting data, training data, classification algorithms, and classification prediction. Training data is defined as attributes for symptoms of the disease (Table 1) and eight types of disease have 23 attributes to be trained (Table 2). Further, this system uses two data storing methods such as MongoDB and stored as files. Frontend also consists of web-based and application-based user interfaces. To identify that the system is working perfectly several experiments were carried out in this study. The technical experiment is done to identify that functionality of the system runs well. The measuring experiment aims to compare the results of the heart rate sensor and temperature sensor are suited to real values. According to the results of temperature sensor average values and real values 0.6-degree Celsius difference was recorded. They assumed that sensor detects temperature properly. The Heartrate sensor value difference compared to the real value is 3.5 beats per minute. Intelligent system experiment focuses on determining symptoms based on early-defined data. Based on the experiment result system accuracy is 90%. Using this system breeders can identify the diseased cow in the early stage and provide them with medical treatments on time[2]Mhatre et. al. have researched to develop a system to measure the milk production of the cattle. This system reads temperature and humidity values, heart rate values, and rumination values to predict the milk yield of the cow in liters per day. DHT11 temperature and humidity sensor, Kg011 heart rate sensor, and ADXL345 absolute motion sensor are used in this system to read data. The sensor values are read using Atmega328p and sent through serial communication to NodeMCU then send these collected data to the ThingSpeak channel. Universal Asynchronous Receiver and Transmitter (UART) is used in serial communication. Network connectivity to NodeMCU is required as it consists of the ESP8266 model. ThingSpeak is an online cloud platform for IoT analysis. Here we can analyze and visualize data with the help of a built-in MATLAB execution system. These four data fields are plotted in ThingSpeak in the form of Graphs and Histograms, then using its MATLAB Analysis Tool system to predict the milk yield. Then it will send to ThingSpeak. Temperature and humidity data were read from the sensor with ±2°C and 5% accuracy. When compared to human cattle has very thick skin. Because of that heart rate sensor used in this system cannot measure the heart rate directly. An amplifier with metal electrodes is used to get a clear picture. The designed amplifier has a gain of 1000 which will amplify weak signals received from electrodes in the range between 1mV- 2mV.

If the predicted milk yield drops below 7 liters per day. Then the system will notify the user through email. In this system, this process will be scheduled to occur once per hour[3].

Minnaert et. al. have done research that is mainly focused on the energy storage solutions for the wearables that are attached to the cow. The power consumption of a wearable is an important obstacle. This will reduce the lifetime of the device and farmers need to regularly change its batteries. As a solution, this paper suggests wirelessly charging the device at the feeding point using inductively coupling. Also, this paper mentioned what is the most preferable energy buffer between rechargeable Li-ion batteries and supercapacitors. In 63s of time period supercapacitors stored 168 J. Rate of energy transfer is not static due to the motions of the cow. At the same amount of time Li-ion battery stored only 100 J. If the system requires high charging time and high energy density it is better to use a hybrid system. This combination will provide a high energy density of Li-Ion batteries and a high-power rate of supercapacitors[4].

Pratama carried out a study in Indonesia for a cattle health monitoring system using the Internet of Things. The system consists of three parts a Collar device to gather data from cattle, a Base station for local server management, and a Web application for analyzing and managing the health of the cattle. Collar capture temperature data, heart rate data, and accelerometer data. Furthermore, it has a 5v solar cell to recharge the battery. Raspberry PI use in this system as a base station that will communicate with collars and get data. Next, it will upload the data to the cloud server. Collected data will analyze and classified through the Machine Learning algorithm into three sections such as normal, less normal, and abnormal. Users will be able to monitor these analyzed data through the web application and keep track of the dairy cow's health[5].

Wang et. al. did a study that describes an early warning system based on cattle activity mainly focusing on this system. The main data source used for this system is activity detection and GPS positioning data. For activity detection, they have used an MPU6050 sensor module with six-axis motion tracking and communication using the I2C protocol. NEO-6M module is used to collect position data. Using the ESP8266 Wi-Fi transmission module the system communicates with the PC. If the system detects any abnormal behavior in the cow built-in alarm buzzer will give a sound reminder to the user. Moreover, the STM32 single-chip microcomputer is used as the core main control unit. The system detects abnormal behaviors in two stages when the steps per hour are less than 2160 and higher than 7200. The system will update the data on the PC and trigger the alarm buzzer if the cow shows abnormal behaviors[6].

Brahim et. al. contributed to a project that is about monitoring the behaviors of the dairy cow. In this system, they have proposed an energy-efficient and reliable method to gather real-time behaviors of dairy cows. Using an accelerometer sensor detects motion activities with the time period. As a power consumption method, this system uses three steps data processing method and a sleep/wake-up method. Components used in this system are Arduino Nano v3.1, MPU-9250 (accelerometer), and nrf24L01 (wireless communication) module. In data processing, three steps are data selection, execution of the classification, and transmission of the result. The main purpose of data selection is to minimize the calculation process and reduce power consumption. The main purpose of this study is to identify the inclination of dairy cows' backs. When an accelerometer detects high inclination system detects it as a transition otherwise the movement is in motion or stationary [7].

P. Chens' study is about developing a cow estrus monitoring system using the Narrow Band IoT (NB-IoT) communication method. The system monitors the body temperature and amount of exercise of the cow frequently. Data will be uploaded to the server using NB-IoT communication and the user can access the data at any time. The system monitors the behaviors of the temperature and exercise amount and determines whether the cow is in the estrus period base on the sudden increase in the data values. Image recognition technology is also used as an auxiliary means of estrus detection. When designing, components are selected to reach the lowest power consumption in the system. STM32 microprocessor is used in this system as a central processing unit. BC-95 module used for NB-IoT communication. ADXL345 accelerometer sensor is used for motion detection and the DS18B20 temperature sensor to gather temperature data[8].

3. Design and Implementation

From the implementation point of view, this system can be divided into three subcategories. They are Smart collar, Web applications, and IoT servers. Fig.1 illustrates the complete system overview.

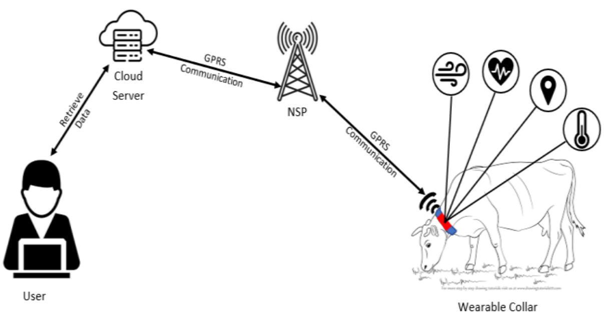

Fig.1 – System Overview

3.1 Smart collar

The main purpose of the Smart collar is to capture information from cattle, such as heart rate, body temperature, the air quality of the surrounding, and the GPS coordinate of its location, then upload it into the cloud server using an inbuilt IoT system. Task-oriented different types of sensors and modules are embedded into the smart collar to capture these data. The Smart collar is mounted on the cow by placing the collar strap around the animal just after the first two legs where the heart is located in the cattle body.

Microprocessor Board – It is based on the ESP32 microprocessor, a single chip that supports Wi-Fi, and Bluetooth (BLE). Antenna and RF baluns, power amplifiers, low-noise amplifiers, filters, and a power management module are also utilized along with the ESP32. The proposed circuitry uses the least amount of space on the printed circuit board as a whole. This circuit board uses TSMC 40nm low power 2.4MHz dual-mode Wi-Fi and Bluetooth chips, which have the best power and RF attributes and are secure, dependable, as well as adaptable to a range of applications [9].

Heart rate sensor – Magene H64 heart rate sensor is used to get heart rate readings of the cattle. The electrode sensors on this heart rate monitor are capable of detecting the electrical activity of the beating heart. Low energy communication is provided by the application that supports Bluetooth BLE protocol compatibility. This also extends the lifetime of the battery. Furthermore, this device is made from strong, abrasion-resistant materials that won't tear, crack, or break. It is IP67 water resistance certified. The user-replaceable CR2032 battery used in the Magene H64 heart rate monitor lasts for an average of 1000 hours [10].

Temperature Sensor – The MLX90614 temperature sensor is chosen to measure the body temperature of the cattle. A non-contact infrared thermometer is present in the MLX90614 for measuring the temperature. The signal conditioning ASIC and the IR-sensitive thermopile detector chip are both incorporated in a single TO-39 container. The MLX90614's low noise amplifier, 17-bit ADC, and potent DSP unit work together to provide the thermometer with significant accuracy and resolution. The thermometer has a digital SMBus output that is factory calibrated and offers full access to the measured temperature across the whole temperature range(s) with a resolution of 0.02°C. The digital output can be set up to use pulse width modulation by the user (PWM). The 10-bit PWM is typically set up with an output resolution of 0.14°C to continually convey the measured temperature in the range of -20 to 120°C [11].

Air Quality Sensor – MQ2 gas sensor is chosen to measure the air quality. MQ2 gas sensor measures the number of gases in the air, including LPG, propane, methane, hydrogen, alcohol, smoke, and carbon monoxide. A gas sensor of the type MQ2 is a metal oxide semiconductor. A voltage divider network included within the sensor is used to determine the gas concentrations in the air. The sensor requires 5V DC power to operate. It is capable of detecting gases with concentrations between 200 and 10,000 pp. The gas sensing component is made primarily of ceramic with an aluminum oxide base coated in tin dioxide and surrounded by a stainless-steel mesh. The sensing element is supported by six interconnecting leads where two are used for heating while the other four are employed to generate output signals [12].

SIM Module – SIM808 board provides cellular GSM and GPRS data in addition to GPS technology for satellite navigation. The board's ultra-low power consumption during sleep mode supports extraordinarily lengthy standby intervals present in the application. Additionally, a LiPo battery-compatible onboard battery charging circuit is present. With 22 tracking and 66 acquisition channels, the GPS receiver is highly sensitive. It also supports assisted GPS (A-GPS) for indoor localization. The board supports 3.3V and 5V logical levels and is controlled by AT commands sent through UART. It comes with a small GSM and GPS antenna; however, it does not require a battery. The board makes use of 2G GSM networks, neither 3G nor LTE [13].

Voltage Regulator – In the proposed design the MLX90614 and MQ2 sensors operating voltage is maintained at 5V whereas ESP32 doesn't have 5V output and consists of 3.3V output. To power up these two sensors directly from a battery, a voltage regulator is used, where an 8.4 V supply is stepped down to 5V. In the process of buck converters, the fixed dc input signal is converted into a different lower-value dc signal [14], [15].

Transistor and Resistor – Due to the direct power supply connection from the battery to MLX90614 and MQ2 sensors, a mechanism should be included to turn off those two sensors when not in use. A MOSFET transistor (IRFZ44N)[16] is used as a switch to turn off the sensors, when not in use. The resistor is used to pull up the signal provide from ESP32 to the gate terminal.

Fig.2 shows the connection diagram between the main processor and the other modules. As shown in Fig.2 all the components in the proposed circuitry are directly communicating with the main processor through wired lines except for the heart rate sensor. The selected heart rate sensor is capable of connecting with the processor through ANT+/Bluetooth 4.0 compatible devices. ESP32 module has an inbuilt Bluetooth system that supports Bluetooth Low Energy protocol (Bluetooth 4.0) [17]. Therefore, the heart rate sensor and the microprocessor are connected through Bluetooth for data communication.

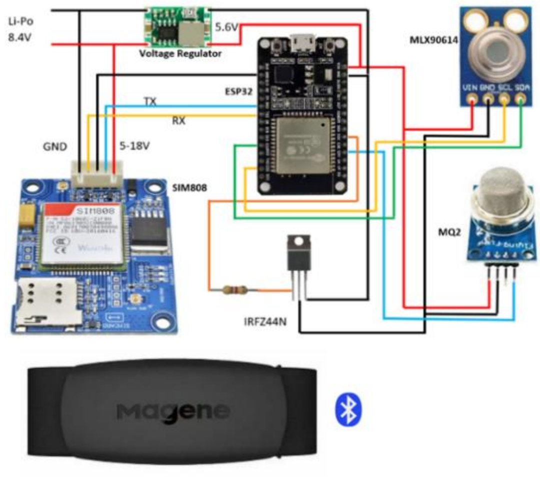

Fig.2 – Connection Diagram

3.1.1 PCB design and device

The Circuit board for the smart collar is designed using DXP Developer software. The schematic diagram of the PCB design is shown in Fig.3, Fig.4, and Fig.5. The printed PCB of the Smart collar is shown in Fig.6.

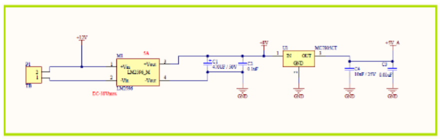

Fig.3 – Schematic of Power Unit

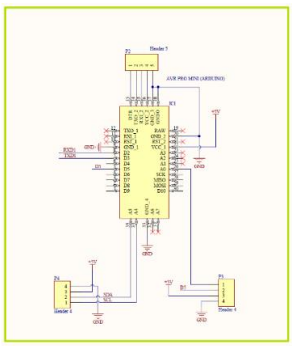

Fig.4 – Schematic of Microprocessor

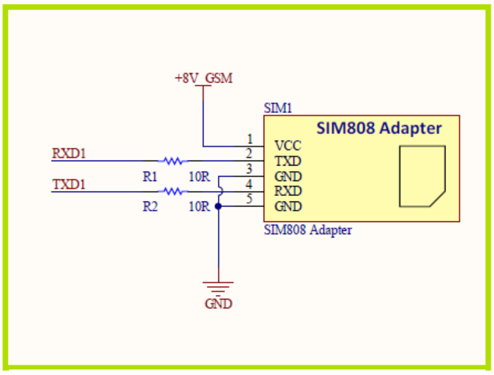

Fig.5 – Schematic of SIM808

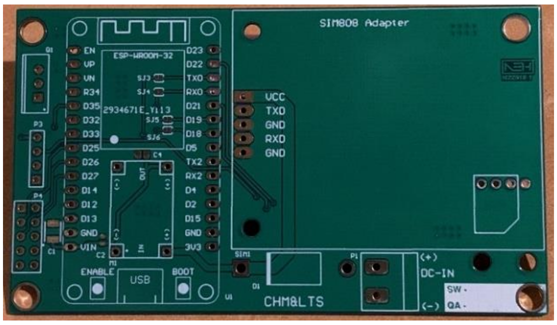

Fig.6 – Printed PCB

The housing of the circuitry boards of the Smart collar is 3D printed using watertight filament. Fig.7 and Fig.8 show the 3D printed model (housing) and Smart collar design.

Fig.7 – 3D Printed Housing

Fig.8 – Smart Collar

3.1.2 Process of a smart collar

Initially, the operating cycle of the smart collar starts with the creation of a Bluetooth Low Energy (BLE) server by the ESP32 microcontroller board to establish the connection with the Magane H64 heart rate sensor. Once the server is created in the ESP32, it will scan for the specific Universal Unique Identifier (UUID) of the specific device. Once the device is located, the connection is established for communication.

If the device cannot be found, the server continues to scan for 15 seconds in each period. If the device is out of reach within this period, the ESP32 resets to begin another round of communication. Once the connection is established the heart rate sensor will not send data directly to the ESP32 until enabling Notify function in the device. To enable notify function, ESP32 needs to transmit enabling code of 0100 to the heart rate monitoring device. Once the notify function is enabled the heart rate reading is transmitted to the ESP32 board. This reading contains a set of information, specifically heartbeat, battery level, position, and signal strength. EPS32 is programmed to filter the data and to obtain only the heart rate data. In general, within 20 seconds, 10 readings are received and the average is considered as the. Once the average is calculated that value is stored in a variable till the transmitted IoT server. After reading the heart rate the ESP32 terminates the BLE server and disconnects from the sensor. After another 10 seconds, the sensor moves to sleep mode until it receives another call from the BLE server.

After getting the heart rate reading ESP32 will move to get data from the MQ2 gas sensor and MLX90614 temperature sensor. The power supply for both of these sensors is controlled by using a MOSFET transistor. To power, the sensors the applied voltage at the Gate terminal will be terminated. Temperature data will be read first. It will give enough time (minimum 6 seconds) to heat the aluminum-oxide-based ceramic, coated with Tin dioxide (sensing element) to get the readings. The temperature sensor will get ten readings with 1 second time intervals and then calculate the average temperature. This will be stored in a variable as cattle body temperature to transmit later. The gas sensor also captures environmental parameters as the temperature sensor and is saved in a variable for later use. Both sensors will calculate the data within 20 second time duration. Once the calculation is done supply voltage at the gate terminal in MOSFET to turn off both sensors.

SIM808 module will move from power down mode to active mode to locate cattle in real-time location and carry out the data transmitting process. First using the built-in function GPS mode will be activated. The microprocessor will read the power supply status for the GPS function. If the GPS is not powered up properly microprocessor will try to power up it for 5 seconds repeatedly. If it fails, then the entire function will be restarted. Once the power is up SIM808 module will try to track the satellite to retrieve longitude and latitude data. When location data is read and stored, the GPS function will terminate.

When all the necessary data collect and store in variables SIM808 will start to build the network connection through GPRS for data communication between the IoT server channel. AT commands will be used to identify the protocols, and network provider, and establish the connection. The collected data will be sent to the IoT server through a link assigning specific variable data to specific fields. After sending the data to the IoT server the network communication will be terminated. Then the SIM808 module will move to a power-down state until it receives a power-up command from the microprocessor.

After completing the data transfer ESP32 microprocessor will move to Hibernation mode for 2 minutes. In this mode entire module functions will be disabled such as ESP32 Core, ULP coprocessor, Wi-Fi and Bluetooth, and Peripherals. A real-time clock is the only function that is enabled. This is used to wake up the microprocessor to begin a new cycle. The total time for one complete cycle execution including the sleep period is around 180 seconds.

3.2 IoT server

In this system, the ThingSpeak IoT server is used to store the data received from Smart collar. Gather, visualize, and analyze real-time data streams in the cloud with the ThingSpeak IoT analytics platform service. Using online services like Twilio and Twitter, you can send alerts, send data to ThingSpeakTM from your devices, and instantly visualize live data. You can create and run MATLAB code with ThingSpeak's MATLAB analytics to carry out preprocessing, visualizations, and analysis. Engineers and scientists can prototype and construct Internet of Things devices with ThingSpeak without setting up servers or creating web applications.

User needs to create a channel in ThingSpeak for their device. This is one-to-one communication for each device it is required to create a different channel. Once the channel is created and synced up with the device the data will be displayed on the application once it is received. Fig.9 will show the interface of a channel that feeds data from Smart collar.

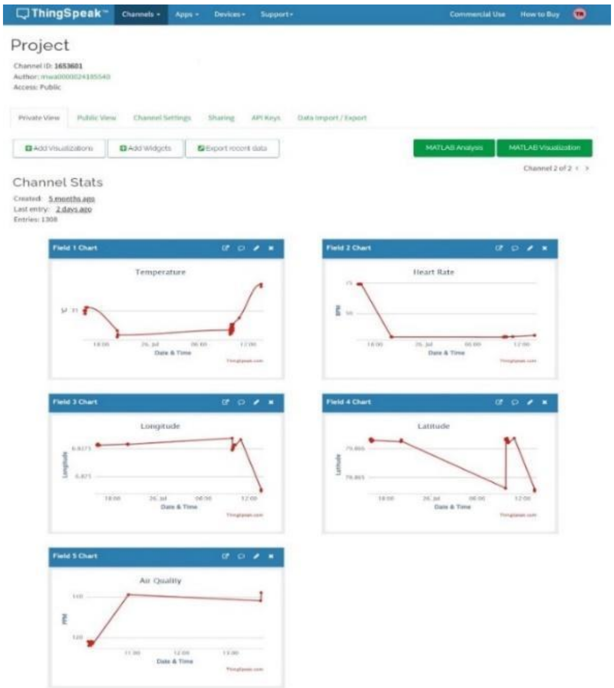

Fig.9 – ThingSpeak Channel

As shown in Fig.9, once the channel is created user will get a unique Channel ID and API Key for the channel. Users can use those IDs and Keys to sync with the device and also add new cattle to the system. A single user can create any number of channels in the ThingSpeak IoT server.

ThingSpeak provides storage space for its channels. This space is up to 8000 data points for a channel. Once it reaches the 8000 range the older data will be removed from the system. Those data can no longer be retrievable. Each field visualized the data points, this data point containing the value, date, and time it received. Threshold values can be added for the fields. If a value exceeds or reduces from threshold vale system can trigger a notification to the user specifying the reason and relevant details.

3.3 Web Application

The web application is used to manage individual cow profiles in the same place. The web application is developed using the Laravel framework. Laravel is an open-source PHP-based application framework. Programmers utilize it voluntarily and it is fully documented. It makes use of the Model-View-Controller (MVC) design, which makes the code clear, comprehensible, and organized and makes website building incredibly simple. When a large group of people collaborates on an application, this style of design is also advantageous. Because Laravel was created using Symfony's components, both web frameworks share the functionalities.

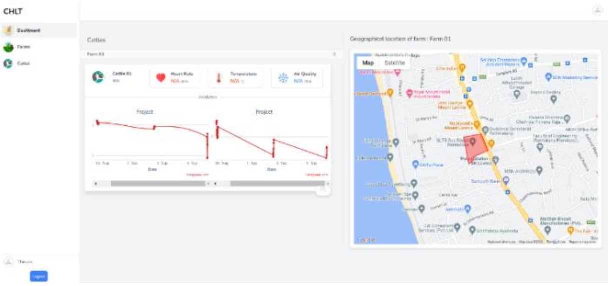

Fig.10 – Application Dashboard

Fig.10 shows the interface of the dashboard.

Initially, a new user account needed to create by the user in the web application using a username, valid email address, and password. Once the user creates the account successfully, the user will redirect to the dashboard of the web application.

The dashboard contains cattle details, farm location, user account, and other functions. Once the user adds a cattle to the application the data will be shown in the application. The graphs can be expanded and minimized whenever the user needs them. Using this application users can manage one or more farms and track data at once. Cattle will be grouped according to the relevant farm. When the user clicks on the cattle he/she wants to see stats the data will be expanded and also the relevant cattle's real-time location will be shown on the map.

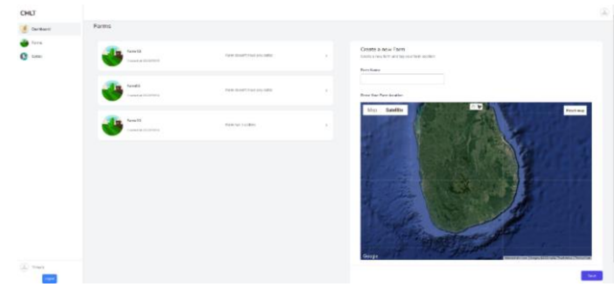

Fig.11 – Farm Tab

When the user clicks on the Farm tab, it will direct to the Farm page shown in the above Fig.11. In here user can add a new farm to the system, or updating a new farm in the system can be done. Users can add a new farm by providing a name and marking the geographical area on the map. Also, an existing farm can be updated by changing its name or editing its geographical area. When the user needs to delete a farm, first the cattle that are assigned to that farm need to be updated to another farm or removed from the system. Then the user can delete the relevant farm.

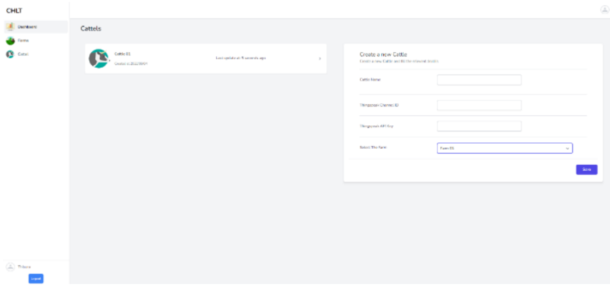

Fig.12 – Cattle Tab

Fig.12 will show the adding new cattle to the system page. Users can add cattle to the system by filling in the required details. Also, the user can update the details of an existing cattle or delete the cattle details from the system. Before adding a new cattle user needs to create a channel for the cattle in the ThingSpeak IoT server. Once the channel is created Channel ID and API key needs to provide to the system to retrieve data from the IoT server. If cattle are added to a new farm that is not in the system it is required to add the farm first before adding cattle details unless selecting the relevant farm that the cattle is belong to is not in the list.

4. Experimental Results

When it comes to portable devices power management is a main concern to increase the lifetime of the device on one charge. Various power-saving methods have been implemented in the Smart collar to increase the lifetime. Starting from the ESP32 microcontroller, the clock speed of the microcontroller is reduced from 240MHz the 80MHz. Below Fig.13 describes the current consumption at different clock speeds.

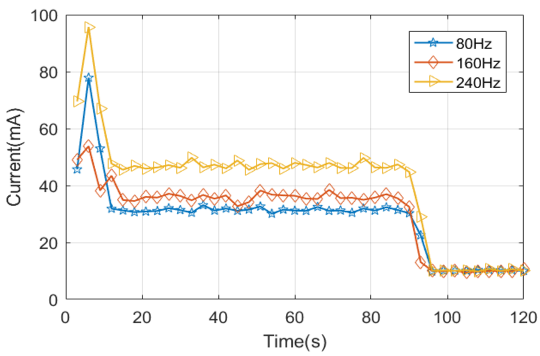

Fig.13 – Current Consumption with Clock Speed

The yellow color line presents the current consumption of the microcontroller at the clock speed of 240MHz. The reason for the peek value at starting point is that the microcontroller creates a BLE server and searches for a heart rate monitor to connect. This process requires a higher amount of current.

The orange color line presents the current consumption of the microcontroller at the clock speed of 160MHz. There is a 12mA to 15mA current difference when compared to 240MHz clock speed.

The blue color line presents the current consumption of the microcontroller at the clock speed of 80MHz. This consumes a low amount of current when compared to the other two clock speeds. When the microcontroller enters Hibernation mode (sleep mode) the current consumption is the same at all three modes because only RTC is active.

Esp32 supports five different types of power modes. They are Active mode, Modem sleep mode, Light sleep mode, Deep-sleep mode, and Hibernation mode. Each mode consumes a different level of current consumption. Below Fig.14 describes the current consumption at each mode.

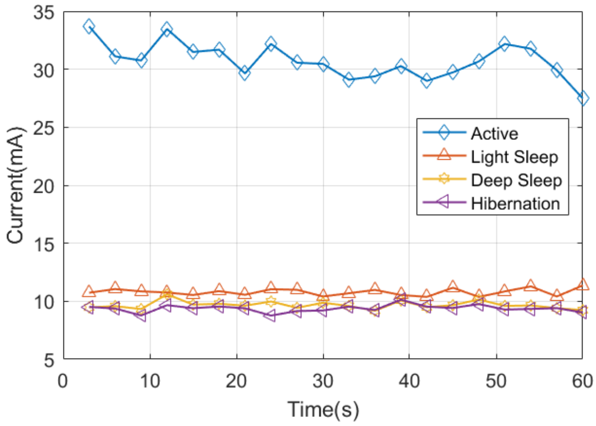

Fig.14 – Microprocessor Current & Voltage Variation in Sleep Modes

The blue color line presents the microprocessor in an active state without processing any commands. All the chip's functionalities are still operational in this mode. Since everything is always running in active mode, including the Wi-Fi module, CPU core, and Bluetooth module, more current is needed to run the chip. Additionally, it has been noted that using Wi-Fi and Bluetooth together might occasionally cause excessive power spikes to emerge.

The orange color line presents the microprocessor in Light sleep mode. The majority of the RAM, the CPU, and the digital peripherals are clock-gated while in light sleep mode. The CPU is put into light sleep mode by turning off its clock pulse, although the RTC and ULP-coprocessor are still in use. Wi-Fi, Bluetooth, and Radio are completely active in this mode. As a result, less energy is used than in active mode.

The yellow color line presents the microprocessor in Deep sleep mode. All digital peripherals and the CPUs are turned off while the microprocessor is in deep sleep mode. The only components of the chip still in use are RTC Peripherals, ULP Coprocessor, RTC Controller, and RTC fast and slow memory.

The purple color line presents the microprocessor in Hibernation mode. In contrast to deep sleep mode, the chip turns off its own 8 MHz oscillator and ULP-coprocessor in hibernation mode. Since the RTC recovery memory is also off, we are unable to preserve any data in hibernation mode. Only one RTC timer (on the slow clock) and a few RTC GPIOs are active because everything else is inactive. They are responsible for waking up the microprocessor from Hibernation mode.

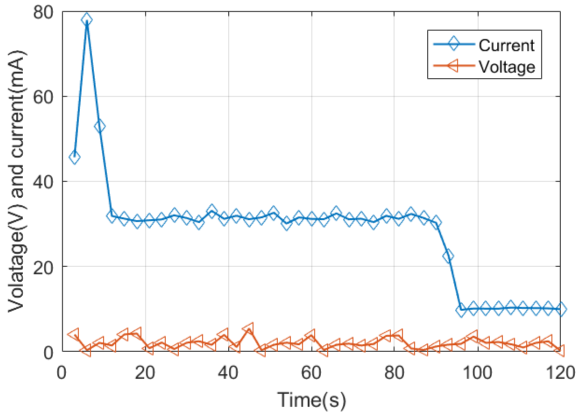

Fig.15 – Microprocessor Current & Voltage Variation

This Fig.15 will show the current and voltage variation in the microprocessor for one complete cycle. The blue color graph describes the current variation and the red color graph describes the voltage variation. This graph is sketched by setting the clock speed at 80MHz. The microprocessor is capable of running much lower clock speeds such as 60MHz, and 40MHz. When setting the frequency lower than 80MHz communication between the microprocessor and SIM808 interrupts. 80MHz is the lowest working frequency for the SIM808 module.

Fig.16 shows the entire system current and voltage consumption for one cycle. The blue color graph presents the current variation and the red color graph presents the voltage variation. When it comes to current there are higher peak values. In the beginning, the system consumes more power to power all units including creating a BLE server and connecting to the heart rate sensor. The peek value between the 20s – 40s is to calculate the readings from the temperature sensor and gas sensor. Gas sensors consume more power between 150mA to 165mA to heat the sensing element in the sensor to get readings. The peek value in the 50s is used by SIM808 to get GPS coordinates. In this processing power the GPS function, search, and lock satellite carry out. The 70s - 90s peek current values are also used by SIM808 to connect to the network for data communication. According to the calculations, the size of one data set is 10kB. SIM808 supports 2G network coverage and it has a speed of 0.1Mbps. The time taken to upload one data set to the IoT server is less than 1 second (10kB/100kbps = 0.8 seconds).

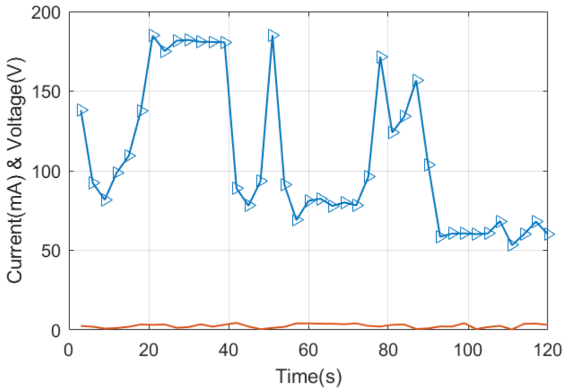

Fig.16 – System Current & Voltage Variation

To determine the lifetime for the Smart collar below parameters measured.

Current consumption of Smart collar per hour = 0.117A

The capacity of the used battery = 20Ah

The lifetime of the device = (20 / 0.117) / 24

= 7.12 Days

This device can fun approximately 7 days from one charge. The following figures are taken during the testing phase.

Fig.17 – Testing Device Reading Realtime

Fig.18 – Cattle Wearing Smart Collar

5. Discussion

The authors intended to add a Smart collar to the Smart collar communication mechanism in further works to construct a sensor network inside the farms. The proposed solution is low-cost and could use in robust environments. The designed sensor node will be expanded to monitor other animals, even though the first deployment was created primarily to support the cow farms. Depending on the demand, the web programs will be upgraded to monitor two or more animal species at once.

6. Conclusion

This paper presents an IoT-based wearable device to monitor the health conditions and locational of cattle in large-scale farms. The proposed design is based on GSM/GPRS communication and can directly communicate the data to the IoT cloud server with mobile base stations without having an intermediate device. Therefore, the proposed wearable collar can be used in robust environments. This system has a design that considers reducing power consumption. Thus, a few power-saving strategies were applied, and evaluated their performance.

References

[1] F. Vannieuwenborg, S. Verbrugge, and D. Colle, "Designing and evaluating a smart cow monitoring system from a techno-economic perspective," *Jt. 13th CTTE 10th C. Conf. Internet Things - Bus. Model. Users, Networks*, vol. 2018-Janua, pp. 1–8, 2017, doi: 10.1109/CTTE.2017.8260982.

[2] Faruq, I. Syarif, A. S. Ahsan, M. Udin Harun Al Rasyid, and Y. P. Pratama, "Health

Monitoring and Early Diseases Detection on Dairy Cow Based on Internet of Things and Intelligent System," *IES 2019 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc.*, pp. 183–188, 2019, doi: 10.1109/ELECSYM.2019.8901527.

[3] V. Mhatre, V. Vispute, N. Mishra, and K. Khandagle, "IoT based health monitoring system for dairy cows," *Proc. 3rd Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2020*, no. Icssit, pp. 820–825, 2020, doi: 10.1109/ICSSIT48917.2020.9214244.

[4] B. Minnaert, B. Thoen, D. Plets, W. Joseph, and N. Stevens, "Optimal energy storage solution for an inductively powered system for dairy cows," *WPTC 2017 - Wirel. Power Transf. Conf.*, 2017, doi: 10.1109/WPT.2017.7953805.

[5] Y. P. Pratama *et al.*, "Designing of a Smart Collar for Dairy Cow Behavior Monitoring with Application Monitoring in Microservices and Internet of Things-Based Systems," *IES 2019 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc.*, pp. 527–533, 2019, doi: 10.1109/ELECSYM.2019.8901676.

[6] Z. Wang, C. Cao, H. Yu, and Y. Liu, "Design and Implementation of Early Warning System Based on Dairy Cattle Activity Detection," *2020 Int. Wirel. Commun. Mob. Comput. IWCMC 2020*, pp. 2186–2189, 2020, doi: 10.1109/IWCMC48107.2020.9148122.

[7] A. Brahim, B. Malika, A. Rachida, L. Mustapha, D. Mehammed, and L. Mourad, "Dairy cows real time behavior monitoring by energy-efficient embedded sensor," *2020 2nd Int. Conf. Embed. Distrib. Syst. EDiS 2020*, pp. 21–26, 2020, doi: 10.1109/EDiS49545.2020.9296432.

[8] P. Chen, "Dairy cow health monitoring system based on NB-IoT communication," *Proc. - 2019 Int. Conf. Electron. Eng. Informatics, EEI 2019*, pp. 393–396, 2019, doi: 10.1109/EEI48997.2019.00091.

[9] "Getting Started with the ESP32 Development Board | Random Nerd Tutorials." https://randomnerdtutorials.com/getting-started-with-esp32/ (accessed Oct. 29, 2022).

[10] "Magene H64 Heart Rate Monitor." https://shop.magenefitness.com/heart-rate-monitors/magene-h64-heart-rate-monitor.html (accessed Oct. 29, 2022).

[11] "Digital Non-Contact Infrared Thermometer (MLX90614) #Melexis." https://www.melexis.com/en/product/MLX90614/Digital-Plug-Play-Infrared-Thermometer-TO-Can (accessed Oct. 29, 2022).

[12] "In-Depth: How MQ2 Gas/Smoke Sensor Works? & Interface it with Arduino." https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/ (accessed Oct. 29, 2022).

[13] "SIM808." https://www.simcom.com/product/SIM808.html (accessed Oct. 29, 2022).

[14] "What is Buck Converter? Operating Principle and Waveform Representation of Buck Converter - Electronics Coach." https://electronicscoach.com/buck-converter.html (accessed Oct. 29, 2022).

[15] "Buck Converters." https://learnabout-electronics.org/PSU/psu31.php (accessed Oct. 29, 2022).

[16] "IRFZ44N Datasheet, PDF - Alldatasheet." https://www.alldatasheet.com/view.jsp?Searchword=Irfz44n&gclid=Cj0KCQjwnvOaBhDTARIsAJf8eVNpWxNtdorj4Hr_k3vOx0EQFL-lBHpvUCApctJ4MsKSB-8uNSfB-LgaAo-KEALw_wcB (accessed Oct. 29, 2022).

[17] "ESP32 BLE Server and Client (Bluetooth Low Energy) | Random Nerd Tutorials." https://randomnerdtutorials.com/esp32-ble-server-client/ (accessed Oct. 29, 2022).

Contribution of individual authors to the creation of a scientific article

KATD Rajapakse: Design and implementation. MWP Maduranga and MB Dissanayake: conceptualized, reviewed, and edited the paper.