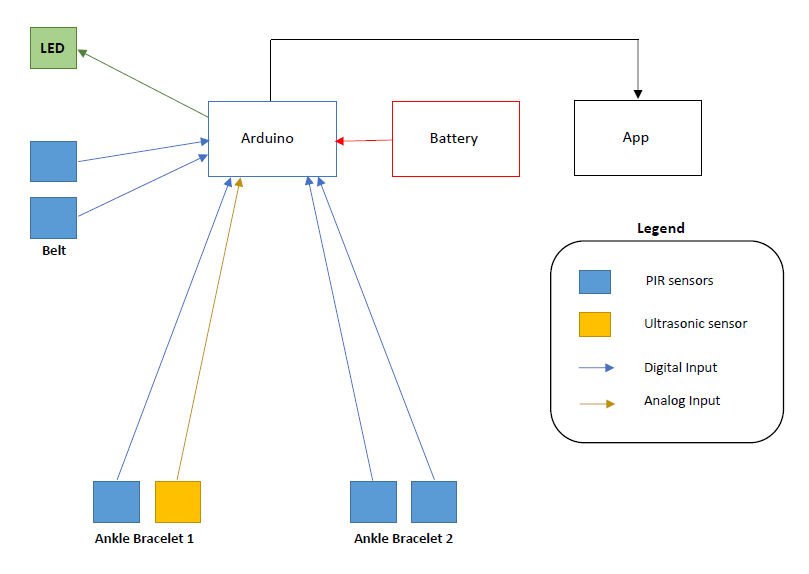
# **2. APPROACH**

GAS provides a way for physical therapists to monitor key aspects of a patient’s gait in between therapy sessions. In particular, GAS is concerned with measuring the average width between the user’s feet, the average time taken for each step, the average time and direction of arm swaying, and the position of the arms relative to the feet. To accomplish this, GAS utilizes Arduino boards paired with Arduino-based ultrasonic motion sensors and Passive Infra Red (PIR) sensors, all of which are located on a belt and two ankle bracelets. GAS stores gait-related information on a memory chip located on the belt, and this data can be accessed through an external app launched on an Android mobile device.



**Figure 1.** Design Overview

## **2.1 Hardware**

GAS includes the following hardware subsystems: the belt, the left ankle bracelet, and the right ankle bracelet. This section lists the numerous hardware components required for each hardware subsystem in GAS. Given that there are no pre-existing tools that record gait-related data, these components were chosen based on functionality, price, and quality.

**2.1.1 Microcontroller**

For the device’s microcontroller, the Arduino Uno Rev3 (referred to as ‘Arduino’ in this document) appears to be the most viable choice. Not only is the Arduino relatively inexpensive and lightweight, but also customizable and easy-to-use [1]. The online community surrounding the Arduino products is centralized around the Arduino website that teaches users how to program their Arduino boards [2], and products are well-documented. With this information in mind, the Arduino is the obvious choice for the microcontroller used in GAS.

**2.1.2 Motion Sensors**

The motion sensors used in this device are responsible for a variety of measurements including: the time it takes for the user to step, the relative position of the user’s feet, the time it takes for the user to swing their arms, the relative position of the user’s arms, and the detection of other people, animals, and objects around the user. PIR sensors are used for these measurements as opposed other types of sensors due to their ability to detect infrared radiation (IR), which allows them to easily detect the presence of an individual [3]. This is the easiest way to detect the presence, or the arms and legs in this case, of the user. Other options that were considered along with PIR sensors were LED sensors and accelerometers. LED sensors are useful since they are able to pinpoint when a specific LED light comes into their field of view [4]. However, if the use of LED sensors was considered over the use of PIR sensors, extra wiring and LED components would have to be placed on the leg opposite the one where the sensors are located. This would not only increase the total manufacturing cost of the device, but also complicate the wiring and component connections. LED sensors also struggle to work well during the day. On the other hand, accelerometers are even worse for determining the required measurement for the device. Accelerometers determine the magnitude and direction of the acceleration they experience via vibration [5]. In order to determine position, the acceleration values from the accelerometer must be integrated twice. In doing this over relatively long periods of time, the error of the device drastically increases [6]. The comparisons among these three types of sensors are summarized in Table 2.1.2.1.

**Table 2.1.2.1 Comparison of Sensors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of Sensor** | **Photoelectric (Able to Detect IR)** | **Works Well in Daylight** | **Accumulated Error** | **Sensitivity to Temperature** |
| PIR | Yes | Yes | Small | High |
| LED | Yes | No | Small | High |
| Accelerometer | No | Yes | Large | N/A |

It can be inferred that the PIR sensor is the optimal sensor when considering the options listed above.

The main options for the type of PIR sensor used in GAS are the following: the HC-SR501 [7], the SEN-13968 [8], and the AMN34111 [9]. General information on the three PIR sensors is listed in Table 2.1.2.2.

**Table 2.1.2.2 PIR Sensors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product** | **Cost** | **Range** | **Measuring Angle (°)** | **Delay Time** |
| HC-SR501 | $2.60 | Up to 3 m to 7 m (adjustable) | 110 | 3 sec to 5 min |
| SEN-13968 | $15.95 | Up to 2 m to 5 m (adjustable) | 110 | 0.4 sec to 7.5 sec |
| AMN34111 | $14.68 | Up to 10m | 110 | 30 sec |

By looking at Table 2.1.2.2, the PIR sensor choice is narrowed down to the HC-SR501 and the SEN-13968. The AMN34111 minimum delay time is too long when compared to the other two sensors, and it has an unnecessarily long range. This PIR’s long-range capabilities cause it to register people, animals, and objects that are relatively far away from the user. The lengthy delay time causes it to miss all the user’s movements within that time period. On the other hand, the SEN-13968 and the HC-SR501 have somewhat similar attributes. The SEN-13968 is expensive when compared to the HC-SR501; however, the extremely short delay time of 0.4 seconds allows the SEN-13968 to successfully record the user’s arm and leg movements without missing any. This is due to the fact that the average human cadence is 100 – 115 steps per minute [10] which is approximately 1.67 - 1.92 steps per second. To measure this, the PIR sensor would need to record the user’s steps every 0.52 - 0.60 seconds on average. With the 0.4 second delay time, the SEN-13968 can detect cadences of up to 150 steps per seconds, which is approximately 30% faster than the upper bound of the average human cadence. Since users are instructed to only walk in the device, the SEN-13968 has more than enough cushion to record all the user’s leg movements. Since the average human arm swing rate closely follows the average human cadence [10], the same logic can be applied to the detection of the user’s arm movements. While the HC-SR501 is much more inexpensive than the SEN-13968, its minimum delay time of 3 seconds makes successfully measuring all the user’s movements very difficult. Therefore, the SEN-13968 is the PIR sensor of choice.

**2.1.3 Distance Sensor**

The distance between the user’s feet must be measured, as it relays key aspects of the user’s gait to the physical therapist. A variety of sensors can be used to accomplish this task, such as infrared sensors, LIDAR sensors, and ultrasonic sensors. Infrared sensors measure distance off of a reflected beam of light; however, they are extremely sensitive to environmental factors [11]. For example, sunlight can poorly affect the performance of this sensor; since patients may be inclined to complete walking sessions outside, infrared sensors are not a viable option. The second option is LIDAR sensors. LIDAR sensors are more advanced than infrared sensors and can even measure 3D structures while maintaining a high accuracy [11]. However, as a result of their advanced technology, they are expensive and can cost hundreds of dollars. On the other hand, ultrasonic sensors are less expensive and do not have the same sensitivity drawbacks experienced by infrared sensors. Ultrasonic sensors emit ultrasonic waves, and the time taken for these waves to be received back at the target determines the distance between the target and another object. Ultrasonic sensors also have a high frequency and high penetrating power, allowing it to easily recognize objects at a high accuracy rate [12]. Based on this, an ultrasonic sensor is an excellent candidate for measuring the average width between an individual’s feet. To do this, this sensor should be informed by the PIR sensors whenever the user’s leg passes the other such that a measurement is recorded by the ultrasonic sensor when the user’s feet are in parallel. The major options for ultrasonic sensors are HC-SR04 [13], PING))) #28015 [14], and US-100 [15]. Important characteristics considered when selecting an ultrasonic sensor are reflected in Table 2.1.3.1.

**Table 2.1.3.1 Ultrasonic Sensors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product** | **Cost** | **Range** | **Measuring Angle (°)** | **Output** |
| HC-SR04 | $3.95 | 2 cm – 400 cm | 30 | Analog |
| PING))) #28015 | $29.99 | 2 cm – 300 cm | Angle not disclosed\* | Analog |
| US-100 | $6.95 | 2 cm – 450 cm | Less than 15 | Analog |

*\*Stated on product website that PING))) has a narrow acceptance angle but does not give specifics*

One of the most important aspects that was considered in the selection process was the range of the sensor. Since the main goal of the sensor is to measure the width in between the user’s legs, the range of the sensor must be adjusted to only measure this width, which is typically 8 to 10 cm for adults [10]. Additionally, the measuring angle is also important. Depending on how low the ankle bracelets are to the ground, the measuring angle, if too large, can cause reflection off the ground, and the ultrasonic sensor would fail to function properly. While HC-SR04 offered the best price and range, the measuring angle was a bit large. However, to counteract this, the ankle bracelets can be moved up the user’s legs (near the shins). By doing this, a proper measurement of the user’s step width is maintained, but any possibility of reflection off the ground is eliminated. Therefore, even if some adjustment to ankle bracelet placement has to be made, HC-SR04 still is the best option for the ultrasonic sensor.

**2.1.4 Subsystem Connection**

There are two main options to consider when it comes to the subsystem connections within the device: wired or wireless. A wired approach to GAS simplifies the design process but complicates the user’s interaction with the device. Connecting the ankle bracelets to the belt via wires means that the ankle bracelets would not need an Arduino or batteries to operate. Instead, the Arduino and batteries are housed in the belt which makes the ankle bracelets lighter and less bulky. Wired connection also allows all the subsystems of GAS to operate synchronously since the delay time in communication with wires is insignificant. Despite these benefits, relatively long wires typically hinder user interaction with a device since the wires can get tangled. On the other hand, a wireless approach to GAS simplifies the user’s interaction with the device but complicates the design process. Connecting the ankle bracelets to the belt via wireless connections means that the user does not have to deal with tangled wires. However, this complicates the design process and makes the ankle bracelets bulkier than necessary. The complication of the design process comes from working with a wireless connection method such as Bluetooth. This causes the design to have two extra battery sources and two extra Arduinos (one for each ankle bracelet). Along with this, there is an extra delay in communication which results in the subsections not working synchronously. Considering these two options, the wired approach appears to be the more viable subsystem connection method for the device.

**2.1.5 Memory Chip**

GAS has to store gait-related data and hold this data until the physical therapist extracts this data. For this to be possible, a memory chip must be used to store all necessary data for a period of time. The two main memory chip options are a Secure Digital (SD) card and an Electrically Erasable Programmable Read-Only Memory (EEPROM) chip. EEPROM, for one, has a smaller memory capacity than a SD card. SD cards are also reliable and user-friendly [16], making them an excellent candidate for this portion of the design. In order to determine the memory capacity for the elected SD card, comparing GAS with another app like MapMyRun is useful. MapMyRun, for example, records its user’s routes, heart rates, location, cadence, and much more while only using around 0.5MB per day [17], [18]. Since GAS records less data from its users than MapMyRun, a 1 GB SD card will be able to successfully capture all gait-related data, even during the worst-case scenario when walking sessions last an hour.

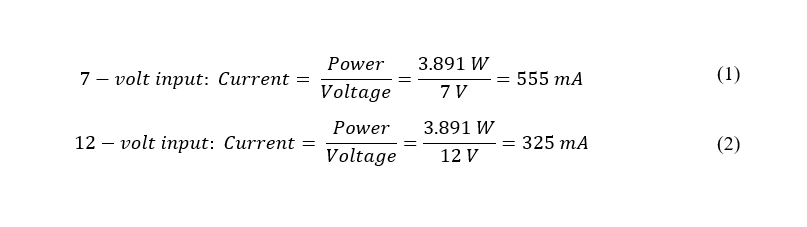
**2.1.6 Battery**

The battery or batteries chosen to power GAS must be able to supply the various components, even at their worst-case specifications. Table 2.1.6.1 lists the expected worst-case power consumption for each component in GAS.

**Table 2.1.6.1 Power Consumption (Worst-Case)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Current (Active)** | **Voltage** | **Power Consumption** |
| Arduino (7 I/O Pins) | 140 mA | 5 V | 0.7 W |
| PIR Sensor  (x5) | 100 mA | 28.75 V | 2.875W |
| Ultrasonic Sensor | 50 mA | 5 V | 0.25 W |
| LED | 20 mA | 3.3 V | 0.066 W |
| **Total** |  |  | 3.891 W |

Calculations performed in Table 2.1.6.1 considered the worst-case scenario with the duty-cycle equal to 1. The total power value of 3.891 W is used to determine the total current of the device. According the Arduino information website, the recommended input voltage is anywhere from 7 to 12 V. Using 7 and 12 V for the upper and lower bounds of the calculations, the recommended currents are the following:



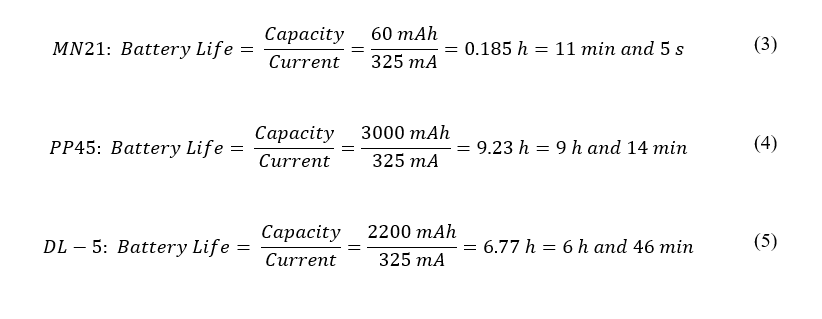
Equation 1 shows that the total current for the device is 555 mA when the input voltage is 7 V, while Equation 2 shows that the total current for the device is 325 mA when the input voltage is 12 V. Since the minimum battery life required for the device is 1 hour, the battery capacity requirement for the 7 V and 12 V inputs are 555 mAh and 325 mAh, respectively. Given these two values, 325mAh appears to be the better condition to meet since it will require a slightly smaller and less expensive battery than the 555 mAh condition. This means that a 12 V battery is the optimal choice for GAS.

When taking types of batteries into consideration, there are three possibilities: MN21 (also referred to as A23) [19], the PP45 [20], and the DL-5 [21]. Table 2.1.6.2 details the specifications of each of these battery types.

**Table 2.1.6.2 Types of Batteries**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Product** | **Cost** | **Volume (cm^3)** | **Weight (g)** | **Output Voltage (V)** | **Capacity (mAh)** |
| MN21/A23 | $2.03 | 2.37 | 8.00 | 12 | 60 |
| PP45 | $44.99 | 258.75 | 579 | 12 | 3000 |
| DL-5 | $3.49 | 43.26 | 54.4 | 6.0 | 2200 |

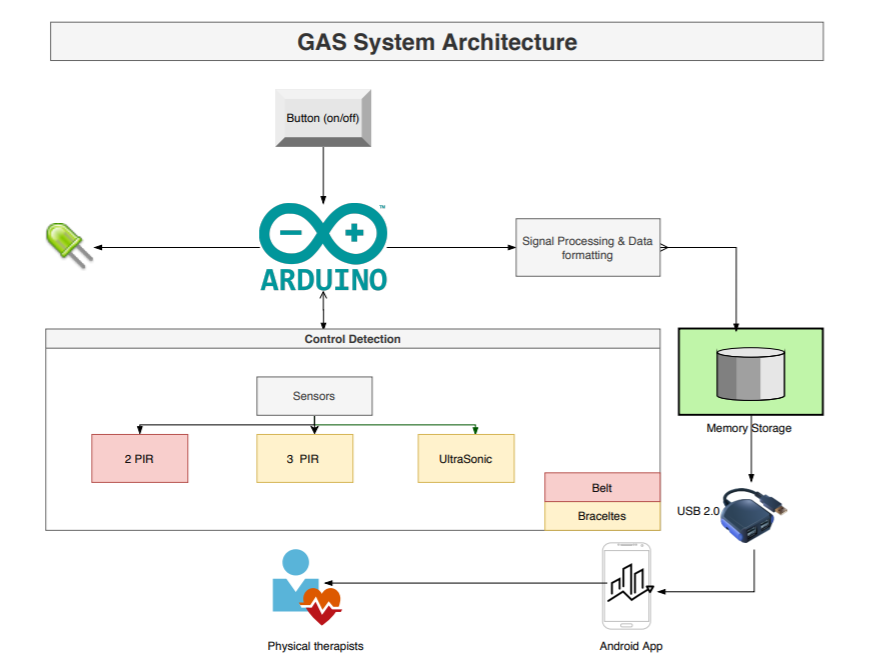
Using the data from Table 2.1.6.2, the following equations provide the battery life of each of the products:



Equation 5 assumes that two DL-5 batteries will be in series providing an input voltage of 12 V to the device. The MN21 and the DL-5 batteries are very inexpensive and relatively lightweight. According to Equation 3, the MN21 only has a battery life of 11 minutes and 5 seconds, which is only 18.5% of the minimum required 1-hour battery life of the device. This vastly diminishes the viability of the MN21. According to Equation 4 and Equation 5, the PP45 and the DL-5 greatly exceed the minimum required 1-hour battery life with battery life values of 9 hours and 14 minutes and 6 hours and 46 minutes, respectively. However, the PP45 is approximately 6.5 times more expensive and takes up roughly 3 times more the amount of space than two of the DL-5s. Given this information, the best battery option for GAS is to place two of the DL-5s in series.

## **2.2 Software**

The software used in GAS includes the Arduino coding, the data storage, and the smartphone app. The Arduino collects the data from the sensors and does any necessary calculations. The Arduino then stores this data until it is transferred to the app. The Arduino will be responsible for the signal processing necessary for data storage. All data collected will be stored in enough memory for one session. The app will be the interface for the physical therapist to view the patient’s data. This information will be visualized in a clear, concise manner. Figure 2.2 visualizes how the components of GAS work together to gather, store, and display information.

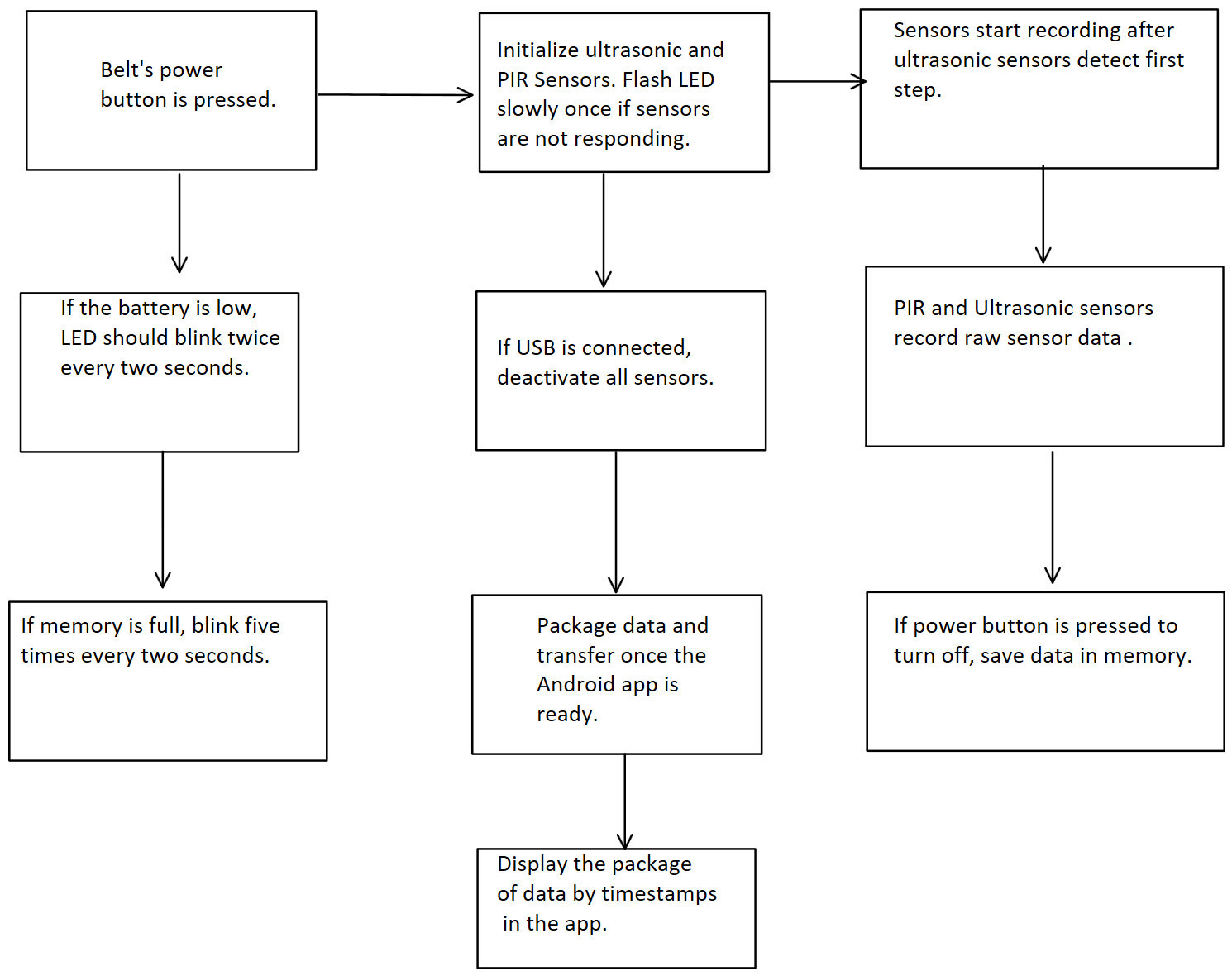


**Figure 2.2** GAS System Architecture

Figure 2.2 shows the overall GAS software system. The inputs to the system come from the sensors measurements, which are stored in the database to be ready to be transferred into the user app.

**2.2.1 Control Detection**

The below flow chart describes the controls states for the Arduino operating system. When the power button is pressed, initialization of all sensors begin. Simultaneously, operating checks of the ultrasonic sensors and the PIR sensors are done to ensure the successful functionality of recording the gait data. When the ultrasonic sensors detect the first step, by calculating the same average distance for 500 microseconds, the PIR sensors can start recording the arm-swings to correlate with the walking data. If the ultrasonic sensors detect a crossing of the user’s feet, it will flag and increment a counter. The detection process is the same for the PIR sensors detecting irregular arm swings.



**Figure 2.2.1** Control Detection Conditions

**2.2.2 Signal processing**

Signal processing involves synchronizing the ultrasonic sensors with the correct frequencies for accurate data capture. Additionally, the raw data from the sensors will be handled by the Arduino software to convert the ultrasonics' analog data to digital data. A built in Arduino function, “pulseIn(),” will then be used to gather the speed of sound, distance, and time. For the PIR, it differs in detection as the software only needs to check if the pin is high. The PIR sensor data will be appended to the ultrasonics’ data to form a calculated table of the average width between the user’s feet, the average time taken for each step, the average time and direction of arm swaying, and the position of the arms relative to the feet.

**2.2.3 Memory storage**

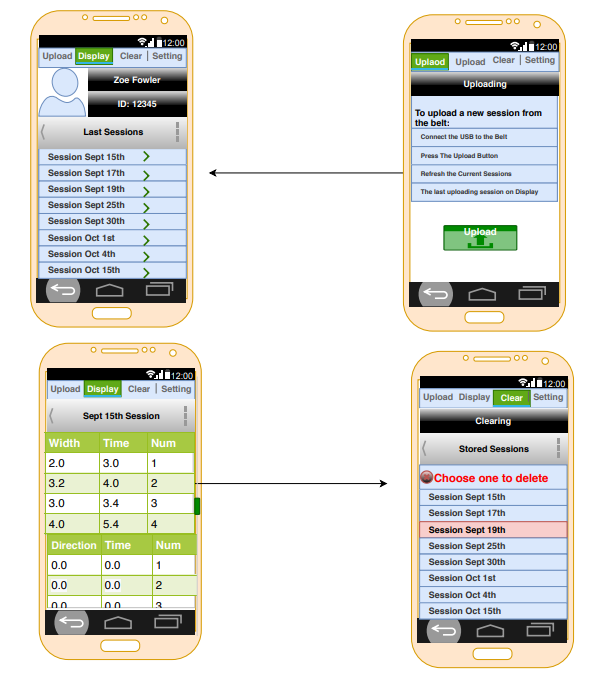
The memory storage is where all data is being stored from the Arduino. Since the physical therapists need to transfer data into the Android app, the data needs to be stored in a specific format which is recognizable by the application. There is more than 0.5MB of data in each session. Therefore, to help the user recognize when the memory storage is full, the LED will blink 5 times every 2 seconds. This visual reminder will keep user’s from continuing a session when no more data can be collected.

**2.2.4 Data Transferring**

The data communication between the app and the Arduino is a wired communication through USB 2.0. First, the user needs to connect the USB cable to the belt, and press the start button to start packaging data. Since there is only one button on the belt, the Arduino needs to interrupt the control detection of if the USB is inserted and deactivate the sensors. The UsbSerial library on the Android is responsible for the serial communication for transferring data over the USB cable [22]. The smartphone will read the external memory of GAS and map the data into the app’s database. Then, the application will synchronously display the results on the user’s interface organized based on the dates of the walking sessions.

**2.2.5 Smartphone Application**

The smartphone application enables the user to upload a new walking session data set from the belt via the USB. The user can view all the current sessions stored in the database of the application organized based on the date of the sessions. Figure 2.2.5 shows the UI design of the application. The backend of the application uses a serial connection library (UsbSerial) that enables the smartphone to read data from the Arduino via the USB [22].



**Figure 2.2.5 GAS’s UI Design**

In Figure 2.2.5, the uploading button is to enable users to upload data, and the display button is to display all the current sessions. The user can also delete any session from the listed sessions through the clear button.

**References:**

[1] “Arduino Uno Rev3,” *Arduino | store.arduino.cc,* 2020. [Online]. Available: https://store.arduino.cc/usa-uno-rev3 [Accessed 16-Sept-2020].

[2] “Getting Started with Arduino UNO,” | *ardunio.cc,* 2018. [Online]. Available: https://www.arduino.cc/en/Guide/ArduinoUno [Accessed 16-Sept-2020].

[3] T. DiCola, “How PIRs Work,” *Adafruit,* 2014. [Online]. Available: https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work [Accessed 16-Sept-2020].

[4] F. Mims III, “How to Use LEDs to Detect Light,” *Make: Community,* 2014. [Online]. Available: https://makezine.com/projects/make-36-boards/how-to-use-leds-to-detect-light/#:~:text=Insert%20the%20fiber%20and%20cement,to%20power%20a%20second%20LED [Accessed 16-Sept-2020].

[5] “Accelerometer,” *Omega.* [Online]. Available:ttps://www.omega.co.uk/prodinfo/accelerometers.html#:~:text=An%20accelerometer%20is%20a%20device,the%20force%20exerted%20upon%20it [Accessed 16-Sept-2020].

[6] “Using Accelerometers to Estimate Position and Velocity,” *CHRobotics.* [Online]. Accessed: http://www.chrobotics.com/library/accel-position-velocity [Accessed 16-Sept-2020].

[7] “HC-SR501 Passive Infrared (PIR) Motion Sensor,” [Online]. Available: https://www.epitran.it/ebayDrive/datasheet/44.pdf [Accessed 17-Sept-2020].

[8] “SparkFun OpenPIR,” *Sparkfun Electronics.* [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/SEN%E2%80%9013968\_Web.pdf [Accessed 17-Sept-2020].

[9] “Passive infrared human detection sensor with built-in amp,” *Panasonic.* [Online]. Available: https://www.alliedelec.com/m/d/9b0187f5be4c4127a102d8868f3222c2.pdf [Accessed 17-Sept-2020].

[10] T. Blackstock, Private Communication, August 2020.

[11] “Types of Distance Sensor and how to select one?” *Seeed Studio,* 2019. [Online]. Available: https://blog.seeedstudio.com/blog/2019/12/23/distance-sensors-types-and-selection-guide/ [Accessed 16-Sept-2020].

[12] “Ultrasonic Sensor working applications and advantages,” *Microcontrollers tutorials & projects,* 2017. [Online]. Available: https://microcontrollerslab.com/ultrasonic-sensor-working-applications-advantages/#:~:text=presence%20detection%20system.-,Advantages%20of%20Ultrasonic%20sensor,the%20external%20or%20deep%20objects.&text=These%20sensors%20have%20greater%20accuracy,and%20depth%20of%20parallel%20surface [Accessed 16-Sept-2020].

[13] “Ultrasonic Ranging Module HC-SR04,” *ElecFreaks.* [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf [Accessed 16-Sept-2020].

[14] “PING))) Ultrasonic Distance Sensor (#28015),” *Parallax.* [Online]. Available: http://www.farnell.com/datasheets/2622313.pdf [Accessed 16-Sept-2020].

[15] “US-100 Ultrasonic Distance Sensor – 3V or 5V Logic,” *Adafruit.* [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Adafruit%20PDFs/4019\_Web.pdf [Accessed: 16-Sept-2020].

[16] “Panasonic SD Cards: Reliable, Secure, and User-Friendly,” *Arrow,* 2018. [Online]. Available: https://www.arrow.com/en/research-and-events/videos/reliable-panasonic-sd-cards [Accessed: 17-Sept-2020].

[17] “Essential Guide to MapMyRun,” *MapMyRun*, 2020. [Online]. Available: https://blog.mapmyrun.com/essential-guide-to-mapmyrun-mvp/#:~:text=MapMyRun%20provides%20users%20with%20the,%2C%20MapMyRun%2C%20MyFitnessPal%20and%20Endomondo [Accessed: 17-Sept-2020].

[18] L. La Porta, “Which Running App is Winning the Data Efficiency Race,” *Wandera*, 2017. [Online]. Available: https://www.wandera.com/running-app-data-analysis/#:~:text=Runtastic%2C%20Runkeeper%20and%20MapMyRun%20all,in%20using%20up%20users'%20data [Accessed: 17-Sept-2020].

[19] “21/23 Battery, 12VDC, Alkaline, Button, 60mAh, PK 2,” *Grainger.* [Online]. Available: https://www.grainger.com/product/DURACELL-21-23-Battery-489D85?ef\_id=EAIaIQobChMIlPr4xbHx6wIVUNyGCh2GSAZlEAAYASAAEgIU1fD\_BwE:G:s&s\_kwcid=AL!2966!3!431945874416!b!!g!!&gucid=N:N:PS:Paid:GGL:CSM-2296:9JMEDM:20500731 [Accessed: 17-Sept-2020].

[20] “12V 3,000 mAh NiMH Battery Pack, PP45,” *Studica.* [Online]. Available: https://www.studica.com/us/en/Studica-Robotics/12v-3-000-mah-nimh-battery-pack-pp45\_1/70018\_1.html?ex\_ref=google\_feed&gclid=EAIaIQobChMIzZDt8dDx6wIVF0qGCh0B3AZ4EAQYAiABEgJHv\_D\_BwE [Accessed: 17-Sept-2020].

[21] “DL-5 Battery,” *Complete Battery Source.* [Online]. Available: https://www.completebatterysource.com/dl-5-battery.php?gclid=EAIaIQobChMI\_YXw7NHx6wIVGEmGCh1lVgdXEAQYAiABEgJYtvD\_BwE [Accessed: 17-Sept-2020].

[22] felHR85, “felHR85/UsbSerial,” *GitHub*. [Online]. Available: https://github.com/felHR85/UsbSerial/. [Accessed: 18-Sep-2020].