ECEN 5823 – Course Project Team Report
4/28/2020
Brian Ibeling

Project Overview

My ECEN 5823 course project (the BLE Motion Detector) shall make use of the SEN-13285 Passive-Infrared motion sensor (PIR) to provide alerts to a user if motion is detected during a window of a user's choosing. A simple use case for this could include detecting if someone has entered a home while the owner is away or sleeping. Another for a future product could be if motion detection would need to be monitored at a remote location for capturing images or collecting data of wildlife remotely. If motion is triggered, a camera or additional sensor could then be enabled for data collection.

Several future components that could be added to this design to enhance the product would be to add a camera that is activated when motion is detected. The camera could then be signaled to record the motion detected and store video to non-volatile memory or a server to be viewed at a later time.

This initial proof of concept design would help to determine if BLE is a viable wireless network and the amount of energy consumed by use of the SEN-13285 PIR motion sensor. The end user may not wish to (or may not be easily able to) perform regular maintenance on a BLE motion detection system, and this proof of concept seeks to provide data as well as an implementation for future product uses.

High Level Requirements

1. The BLE Motion Detector system shall use BLE as the primary method of communication between the server motion detection device and any bonded client connection(s).
2. The BLE Motion Detector system Server device shall be able to detect motion within 10 feet of sensor placement.
3. The BLE Motion Detector system Server device shall allow the end-user to schedule a time range that motion detection should be monitored.
4. The BLE Motion Detector system Server device shall include custom Motion Detected read services which provides service data to include (a) time of last motion detect event and (b) the current motion detection schedule.
5. The BLE Motion Detector system Server device shall include custom write service to update the Motion Detection Schedule Start time and Stop time in hour increments.
6. The BLE Motion Detector system Server device shall provide service information to securely bonded clients only.
7. The BLE Motion Detector system Server device shall display current system state, current time, time of last motion detected, and current motion detection schedule on the Server LCD.
8. The BLE Motion Detector system Server device shall store the time of the last motion detection event captured on persistent non-volatile memory on-board the Server device.
9. The BLE Motion Detector system Server device shall store the last set detection schedule on persistent non-volatile memory on-board the Server device.
10. The BLE Motion Detector system Server device shall provide a secure method for device firmware upgrades via Over The Air (OTA) updates.
11. The BLE Motion Detector system Server device shall provide a client-server bonding method that prevents MITM (Man-In-The-Middle) attacks from occurring.
12. The BLE Motion Detector system Server device shall be in Energy Mode 3 (EM3) while the device is not actively collecting data for detected motion.

13. The BLE Motion Detector system Server shall enable the primary application software when system power is provided or a reset event occurs.
14. The BLE Motion Detector system Server device shall send a signal or alert to enable the data collection subsystem or device attached each time motion has been detected.
15. The BLE Motion Detector system Server device shall send a signal or alert to a user each time motion has been detected.
16. The BLE Motion Detector system Server device shall allow the end-user to schedule a time range that motion detection should be monitored.

High Level Design

The BLE Motion Detector system shall use a Silicon Labs Blue Gecko as the BLE server device's hardware platform. This will leverage Silicon Labs' system API and drivers to access necessary energy modes and features provided by the Blue Gecko hardware. Software developed for this server implementation will run within an interrupt and event driven scheduler that is shared between BLE and system events.

Software state machines will be used for monitoring and determining system operating sequencing for both the BLE connection state and motion detection events. BLE connections will follow BLE standard server-client connections and secure DisplayYesNo bonding procedures, and advertise each time the system boots, or a client is disconnected. A state machine will also be implemented for detecting motion, fluctuating between not detecting and detecting motion. In the not detection motion state, no data is logged and the system is in a low-power sleep mode if BLE events are not being handled. In the detecting motion state, motion is detected and an interrupt is triggered to capture the event and current system time. This is saved to system Flash memory and the LCD display updated with the associated information.

Three services will be exposed once the server has a bonded connection to a client. The first will be a motion detected service, which will update and provide indications to the bonded client to the timestamp of the last motion event captured, as well as the current detection schedule last set by the user. The time of each detected motion event will be logged and saved to persistent memory.
A write service will also be available to set the schedule start and stop times in hourly increments. Each update to the detection schedule will be saved to persistent flash memory, and is loaded each time the system is booted. By default, this schedule will be set to detect motion during a full 24-hour period.

The system shall also allow for secure firmware updates via Over-The-Air (OTA) to reload the application image onto a device from a Bluetooth connected client smartphone running the Silicon Labs software.

To simulate an end-product for the POC demo, an on-board LED will be used to mimic a camera being enabled to capture images or video. This will be enabled when motion has been detected by the system, and will help to simulate/visualize when images or video is being captured.

For the system prototype, a simple smartphone app provided by Silicon Labs (such as the EFR Wireless app or the Wireless Gecko app) to confirm that services are advertising properly and that motion detection is being reported when it occurs.

Figure 1 below shows a block diagram of the system detailed above.

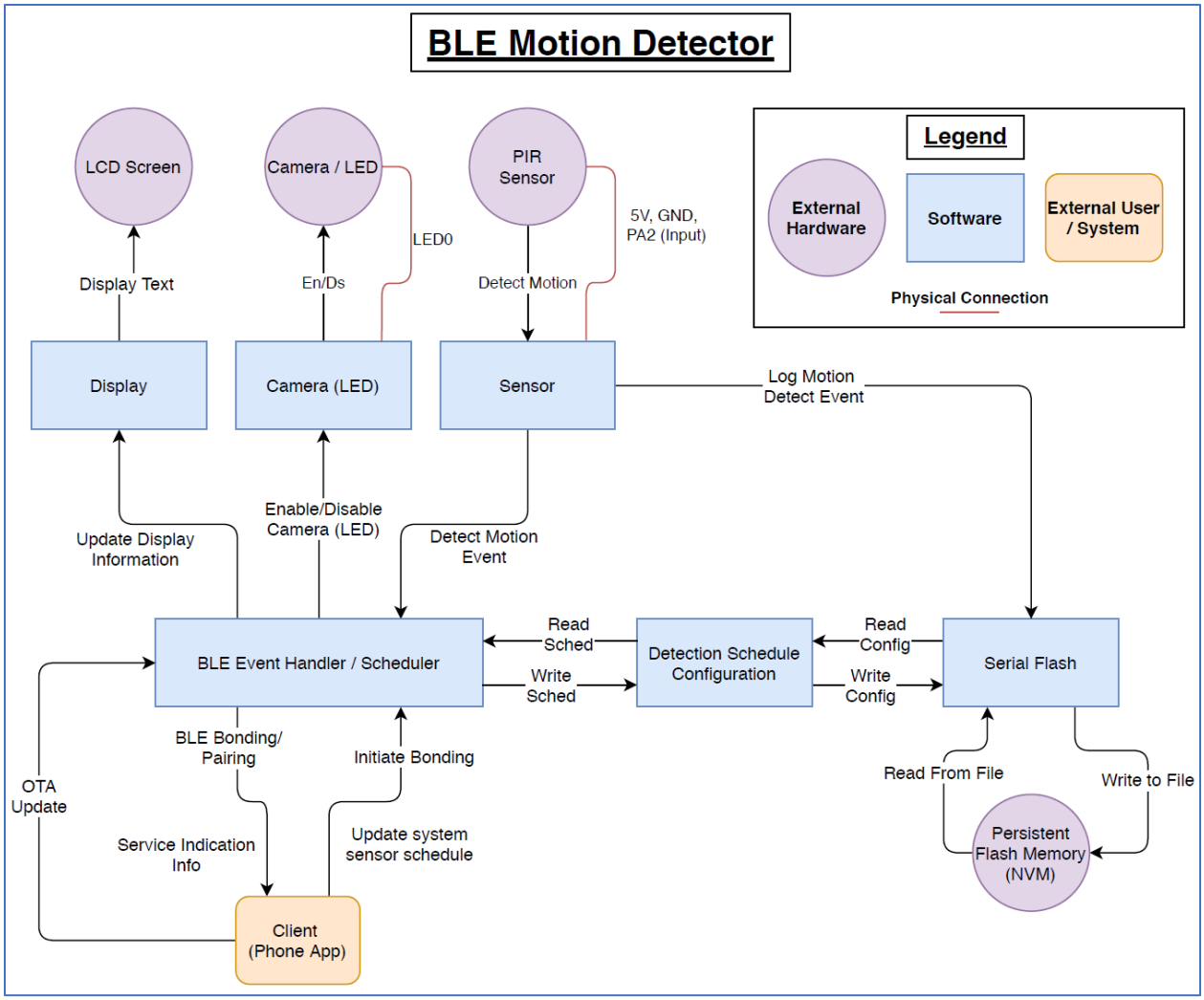

Figure 1: BLE Motion Detect System Diagram

The table below in Table 1 describes the PIR Motion Sensor being used of this POC system.

Table 1: Motion Detect Sensor

Motion Detect Sensor	
Sensor	Passive Infrared (PIR) Motion Sensor
Interface Method	3-pin Analog Sense (active low)
Sample Rate	Interrupt driven
Sensor Range	10 Feet
Link	https://www.sparkfun.com/products/13285

Subcomponent Summary

Sensor

The sensor subsystem will provide an interface for the PIR motion detection sensor, sampling the analog input of the device. This info will be provided to the BLE Event Handler and Scheduler subsystem to determine if the camera subsystem should be enabled or disabled.

The state of the sensor subsystem will change based on if motion is detected or not by the sensor. This state will then be provided the LCD to be displayed to the user and be saved to system memory with the timestamp of the event along with the relative distance of motion.

It should also be noted that the device requires a 1-2 second delay upon startup to allow the device to calibrate, and will be taken into account when developing this application.

BLE Event Handler/Scheduler

The BLE Event Handling and Scheduler subsystem is the primary operating monitor of the application. This will handle all BLE events as well as user defined events. Triggered interrupt will set the corresponding event flag, which is then triggered and handled within this event handler. Pairing and connecting to BLE clients are handling here, as well as the passing of service indications back to the client. System state will also be closely monitored by the subsystem, with motion detect and BLE bonding states updated as needed.

Serial Flash

The Serial Flash subsystem will handle read/write events with on-board persistent flash memory. This will also data to be retrieved after power cycle and power loss events.

Two primary pieces of information will be stored in flash memory – logging data and the user set sensor sample schedule. Logging data will be focused on the last motion detect event detected, with the time of the last event captured. The motion detection schedule will be a user-specified schedule for which sensor sampling should occur. This will allow the user to specify when they wish to monitor and be alerted of detected motion events. This information will load the last config set each time the system is powered or reset. In the event no config is found in memory, a default schedule will be specified to alert motion detection during all hours of the day.

Display

The display subsystem will manage information displayed the BLE Blue Gecko server LCD screen. These will showcase the current BLE connection state of the server with a client (disconnected, connecting, bonding, bonded, advertising). Additionally, the current motion detected state will be display as well, showing if motion has been detected in the last 1 second interval and the approximate distance of the motion detected.

Firmware Update

The Firmware Update subsystem will handle updates received from the end user for the application software. Additional, this will be used to receive the sensor sample schedule (See Memory/Logging above) and pass the necessary information along to the Config subsystem for writing data to flash memory.

Detection Schedule

The Detection Schedule subsystem will handle checking if a motion event is within a user-set range of hours for motion detection. Additional, this will provide methods for setting the detection schedule when a user specifies a new start or stop time from the client phone application.

Project Team Members

Brian Ibeling

Validation and Test Plan

The course project Validation and Test Plan is provided at the following link: https://drive.google.com/open?id=1cl_kAOV65JaB7zqjiRuHWzLagP5wgar-. This is a link to the excel spreadsheet uploaded to the BLEMotionDetector Google Drive folder for this project to a file titled "Course_Project_Validation_Plan.xlsx".