**Project Logbook**

**Title:** ESP32-Based Security Camera with Motion Detection and BLE Server  
**Timeline:** September – November 2024  
**Platform:** Arduino / ESP32  
**Key Technologies:** ESP32-CAM, PIR Sensor, BLE (Bluetooth Low Energy), MicroSD Storage, Deep Sleep, Arduino IDE

**September 2024 – Project Planning and Early Prototyping**

**1–3 September**

* Outlined the initial idea: a small, standalone security system using ESP32-CAM.
* Defined key requirements: motion-triggered image capture, BLE alert system, low power draw.
* Decided to use BLE over Wi-Fi for local, low-energy notifications.

**4–6 September**

* Researched and compared ESP32-CAM capabilities with other microcontrollers.
* Selected components: ESP32-CAM board, PIR motion sensor (HC-SR501), AMS1117 regulator, MicroSD card reader (built-in), and 5V power supply.

**7–10 September**

* Sketched a basic wiring diagram on paper and prepared a breadboard layout.
* Connected PIR sensor to ESP32 and wrote a basic sketch to print motion detection events to the Serial Monitor.

**11–14 September**

* Verified PIR sensor response time and detection distance.
* Tested camera module functionality separately: captured and saved static images to SD card using onboard camera.

**15–20 September**

* Experimented with basic BLE examples using the Arduino BLE library.
* Set up a simple GATT server that broadcasts a single service and test characteristic.
* Began planning for BLE notification triggers tied to PIR sensor events.

**October 2024 – Core Development and System Integration**

**1–5 October**

* Implemented interrupt-driven motion detection: PIR sensor triggers a function that captures an image.
* Saved image files to MicroSD card with timestamp-based filenames.
* Ensured files were readable and correctly stored by removing the card and inspecting on PC.

**6–10 October**

* Developed BLE server with a motion notification characteristic.
* Integrated BLE server into the existing sketch and tested stability of BLE broadcasting.
* Successfully sent notification flags to BLE client (tested via nRF Connect app).

**11–15 October**

* Implemented ESP32 deep sleep mode to conserve power.
* Configured wake-up via GPIO interrupt from the PIR sensor.
* Tested boot time and reliability after repeated sleep/wake cycles.

**16–20 October**

* Combined camera capture, BLE server setup, and motion detection into a single integrated sketch.
* Resolved timing issues: BLE server needed to reinitialise properly after deep sleep cycles.
* Used delay management to avoid missed triggers or SD card write failures.

**21–25 October**

* Created a prototype enclosure using cardboard and plastic to mount the ESP32-CAM and sensor.
* Positioned the device in different locations indoors for field testing.
* Observed trigger accuracy and light conditions affecting image quality.

**26–31 October**

* Improved BLE characteristic by adding custom service UUIDs and reliable notification logic.
* Added LED indicator to show when an image was being captured or motion detected (debugging aid).
* Logged image timestamps and motion events to serial output for further debugging.

**November 2024 – Testing, Optimisation, and Documentation**

**1–5 November**

* Performed a full-week test: left the device active near a hallway.
* Collected image logs and BLE notifications via mobile app.
* Analysed false positives (triggered by pets, sunlight shifts) and adjusted PIR sensitivity.

**6–10 November**

* Cleaned and refactored code:
  + Removed unused libraries.
  + Added inline comments and structured functions.
  + Handled edge cases like SD card not inserted or BLE disconnects.

**11–14 November**

* Created circuit diagram using Fritzing.
* Compiled a comprehensive README with:
  + Overview of functionality
  + Hardware list and assembly instructions
  + How to flash and test the sketch
  + BLE client usage instructions

**15–17 November**

* Recorded a short demonstration video showing the camera capturing images and BLE alerts received on a phone.
* Created and annotated demo images to showcase system output.

**18–20 November**

* Finalised GitHub repository:
  + Pushed final version of Arduino sketch.
  + Included all diagrams, sample image files, and a link to the demo video.
  + Wrote up lessons learned and ideas for future improvements (e.g., battery-powered version, Wi-Fi integration, external storage).

**Summary**

By the end of the project:

* A fully functional security camera system was built using ESP32-CAM.
* It detects motion via PIR, captures images to MicroSD, and sends BLE alerts to paired devices.
* Power consumption was reduced using sleep modes.
* The project was fully documented and released as open-source on GitHub for demonstration and reuse.