Week 9 Lab

CC2511

Background

You are designing an embedded system to handle confidential and valuable data. However, you are acting dishonestly. Your device will be sold to a particular target whose data you would like to steal. The data are worth a lot of money to you, but it is important that your ruse is not detected. If the target notices that your embedded system is exfiltrating data then your evil scheme will be exposed.

Your target is technically competent and would notice a radio antenna or a length of wire running back to your lab (or evil lair). You therefore need to be creative, and use something non-obvious as a communication channel.

You think carefully about how to smuggle out data, and after some time, you are struck with inspiration. *All devices have a status light.* No one would be suspicious of a happy blinking light. It is totally normal that a flashing light indicates some activity or status of your system.

A human is unlikely to notice the difference between a 40 ms pulse of light and a 50 ms pulse of light. You decide to implement a communication scheme whereby you will modulate the duration of the light pulse in order to transmit information.

Your surreptitious communication scheme will operate according to Figure 1, below.

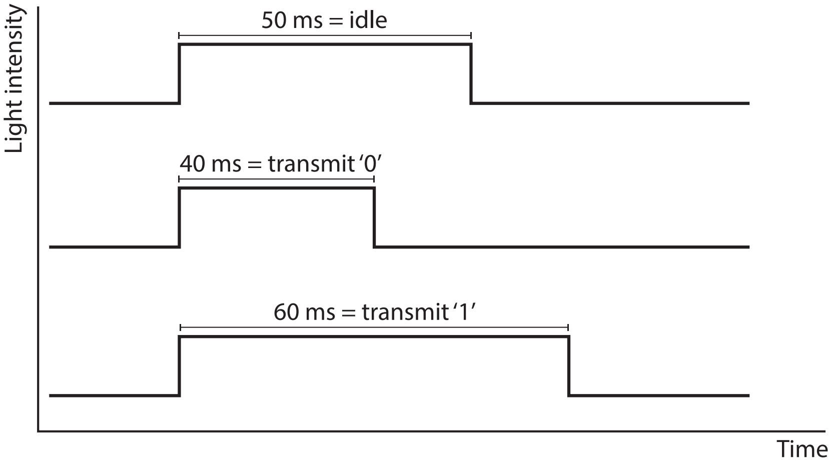


Figure 1: Proposed communication scheme using flashes of light.

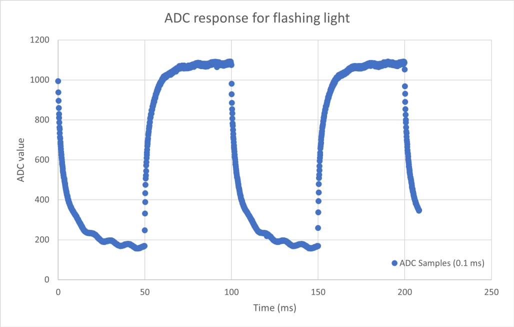
To read the data, you will arrange for a receiver device to be placed nearby to your system so it can observe the light flashes. Since the sensitive device is "air gapped", your target will assume that it is secure and no communication is possible. In this task, you will show that that assumption is naive.

Your task

Working in pairs with another student, implement a transmitter and receiver for the scheme in Figure 1. You should aim to work collaboratively in a "pair programming" style where you take turns coding while the other watches and talks about the task with you.

Specifically:

1. Implement a transmitter that continuously generates flashes of white light by controlling all 3 of the LEDs on your development board. You might begin by always generating 50 ms pulses, or you might slow it down to 500 ms for the purposes of easier measurement during development.
   - Hint: the sleep_ms() function can be used to set the delay in milliseconds between actions.
2. On a second CC2511 Pico Development board, observe the changes in light intensity picked up by the sensor when it is held near to the flashing transmitter. You might try one of these approaches:
   (a) Connect an oscilloscope to the light sensor and observe the change in voltage when you hold the sensor close to the transmitter[1], or
   (b) Write software code to periodically print out the value returned by the sensor. You will need to slow down the blinking so that you can explore the difference between the two transmitter states.
3. Continuing to work on the receiver, write code to detect rising and falling edges of light intensity. The easiest way to do this is by comparing the current and previous ADC measurement. If the difference is sufficiently large and positive, you have a rising edge. If the difference is sufficiently large and negative, you have a falling edge.



[1] The light dependent resistor's pins are exposed on the PCB so it is easy to attach an oscilloscope probe.

4. Measure the time that elapses between rising and falling edges. Print out this time to the terminal, e.g. “Pulse width: 50 ms”.

5. Configure your transmitter to send pulses of different widths and confirm that you are able to distinguish them on the receiver side.

Optional extension

1. Implement the ability to transmit text between boards. For example, you might type into the serial port on one end and have it transmitted by light pulse to the other board, and displayed on its serial terminal.

   You will need to transmit a single bit at a time. You can detect the least significant bit in a byte using

   ```
   if (msg & 1) { /* LSB is 1 */ }
   ```

   and then shift all the bits so that the next bit to transmit becomes the least significant bit in the variable:

   ```
   msg = msg >> 1;
   ```

2. Discuss with your classmates and/or prac tutor:

   (a) What are the implications of placing trust in complex machines such as microcontrollers, and

   (b) How could you defend yourself against similar attacks?

Assessment

To finish this lab, demonstrate the following to your tutor:

- A transmitter that generates light flashes with different pulse widths.

- A receiver that detects the pulse width or decodes the messages.

- Show your prac tutor your GitHub webpage where your source code is uploaded.