Exposing Security Vulnerabilities of Infrared Transmission

The Team

Cooper Hammond

- PCB Design

Kobe Prior

- Eavesdrop Software

Blane Miller

- OLED Software

OUTLINE

1

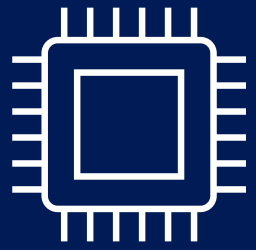
LIVE DEMO

1

LIVE DEMO

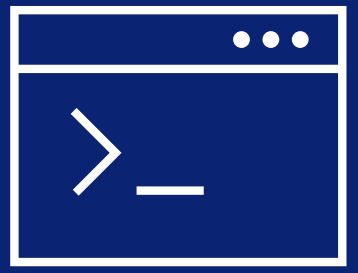
2

PCB



3

SOFTWARE

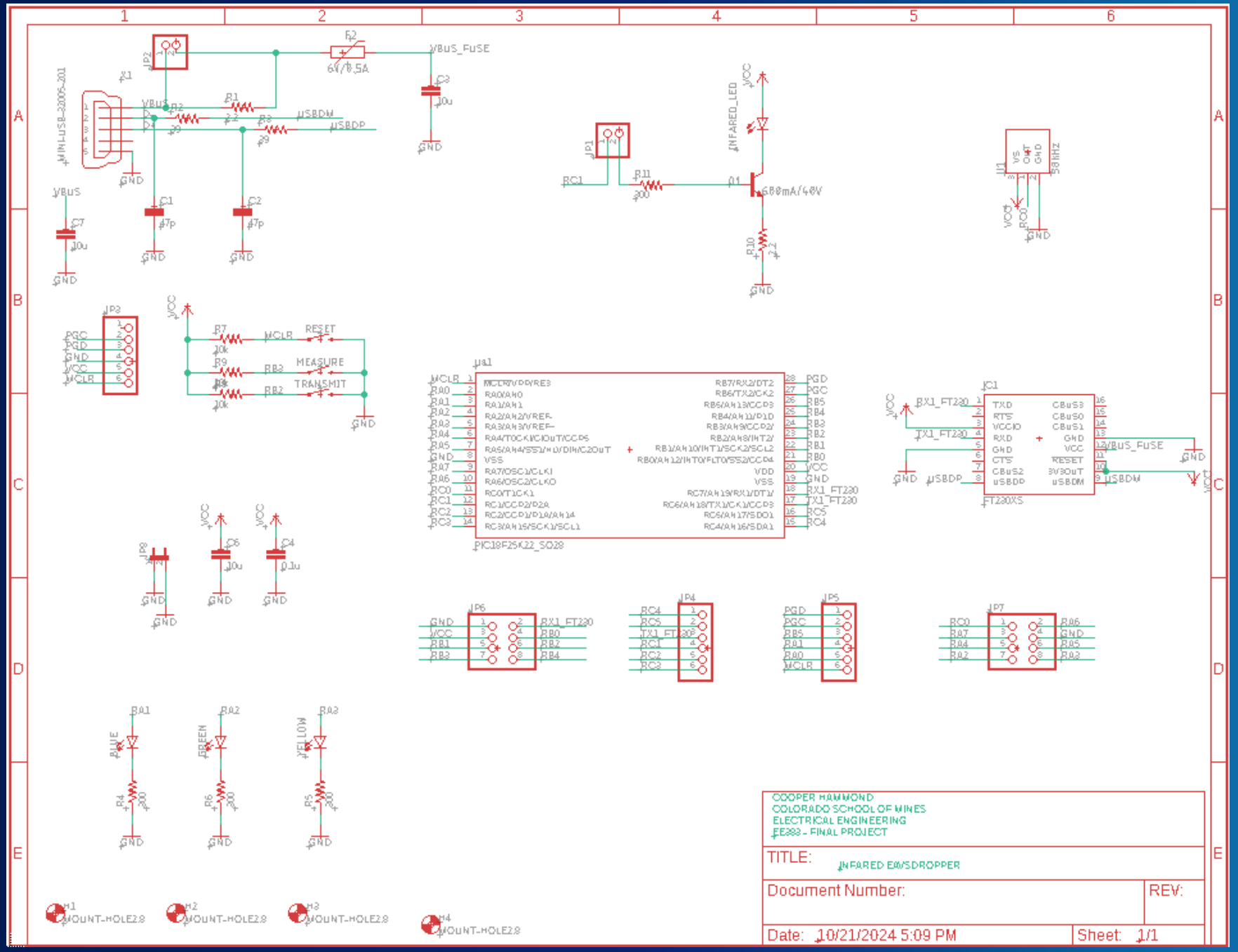


4

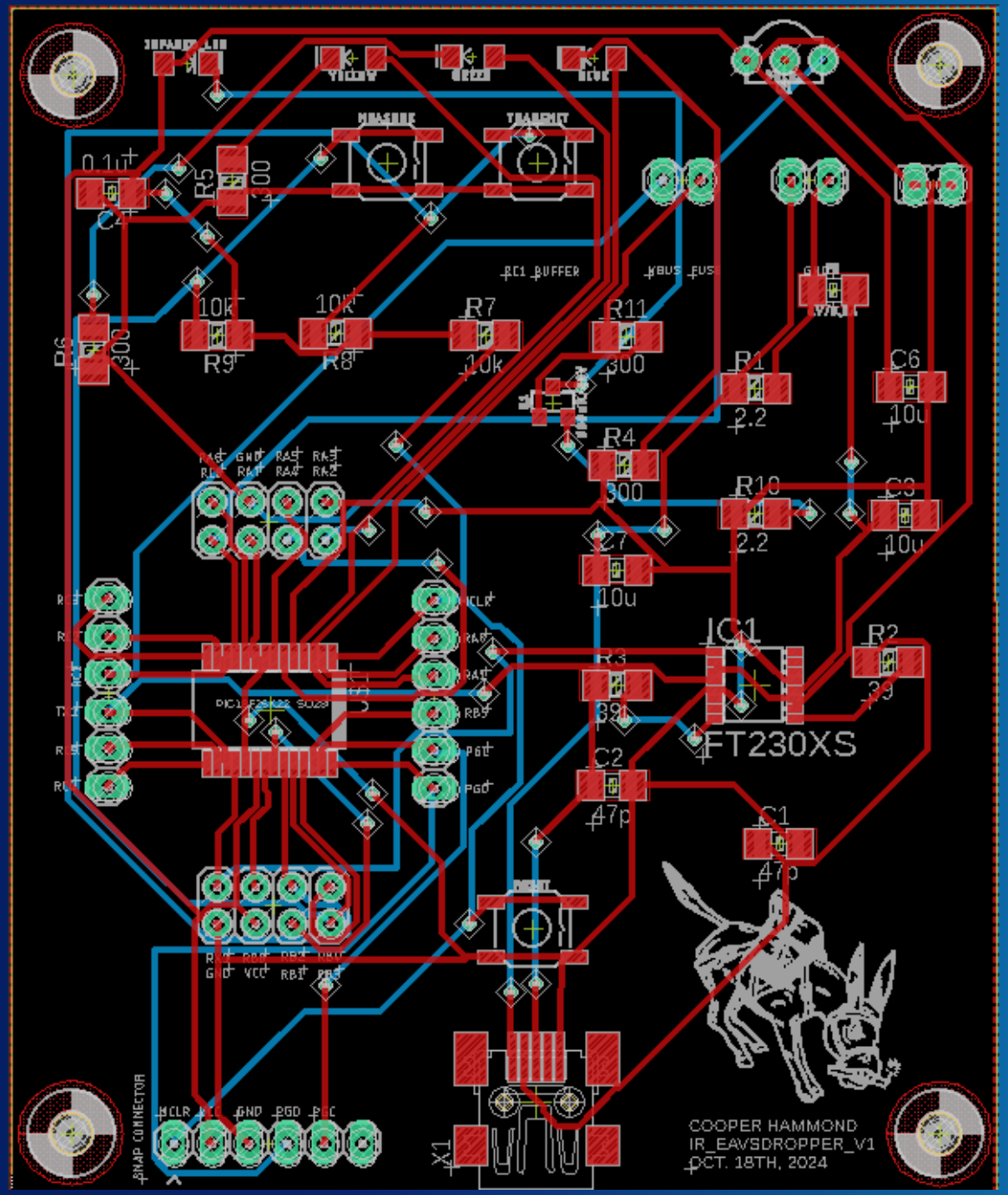
C.B.K. Protocol

LIVE DEMONSTRATION

Research and Develop Schematic

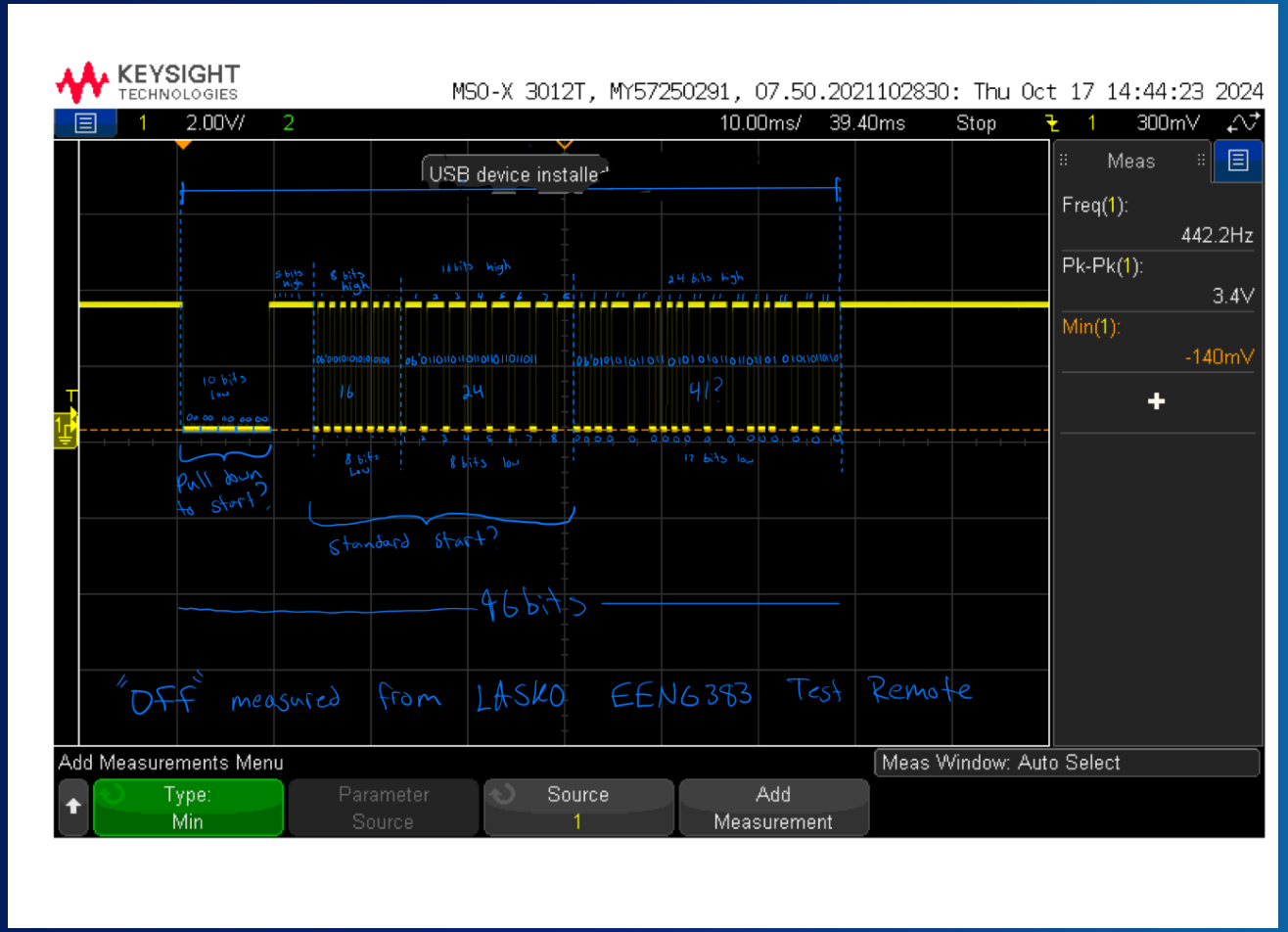


Research and Develop Board Design



Eavesdrop Software

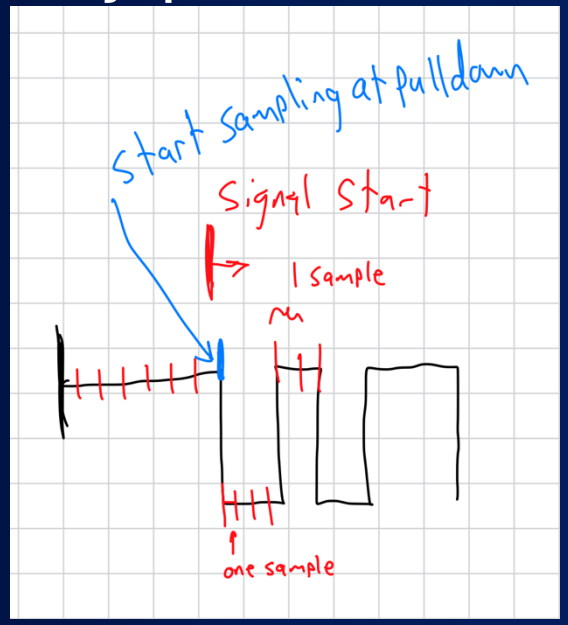
- Goal: Record the output of the IR decoder, and replicate that signal using PWM for IR LED
- Challenges: Data constraints - we can only hold so much data from the IR decoder



Key Discoveries

Strategically Sample

- Start Sampling at pull-down
- Sample at Nyquist rate (2xBaud)



Data Structure

- To store the samples, we use an uint8_t array called receive buffer
- Each sample is a 0 or a 1, to store lots of samples we access each byte in the buffer and assign each bit in that byte
- More on the next slides

Key Helper Functions to use data structure

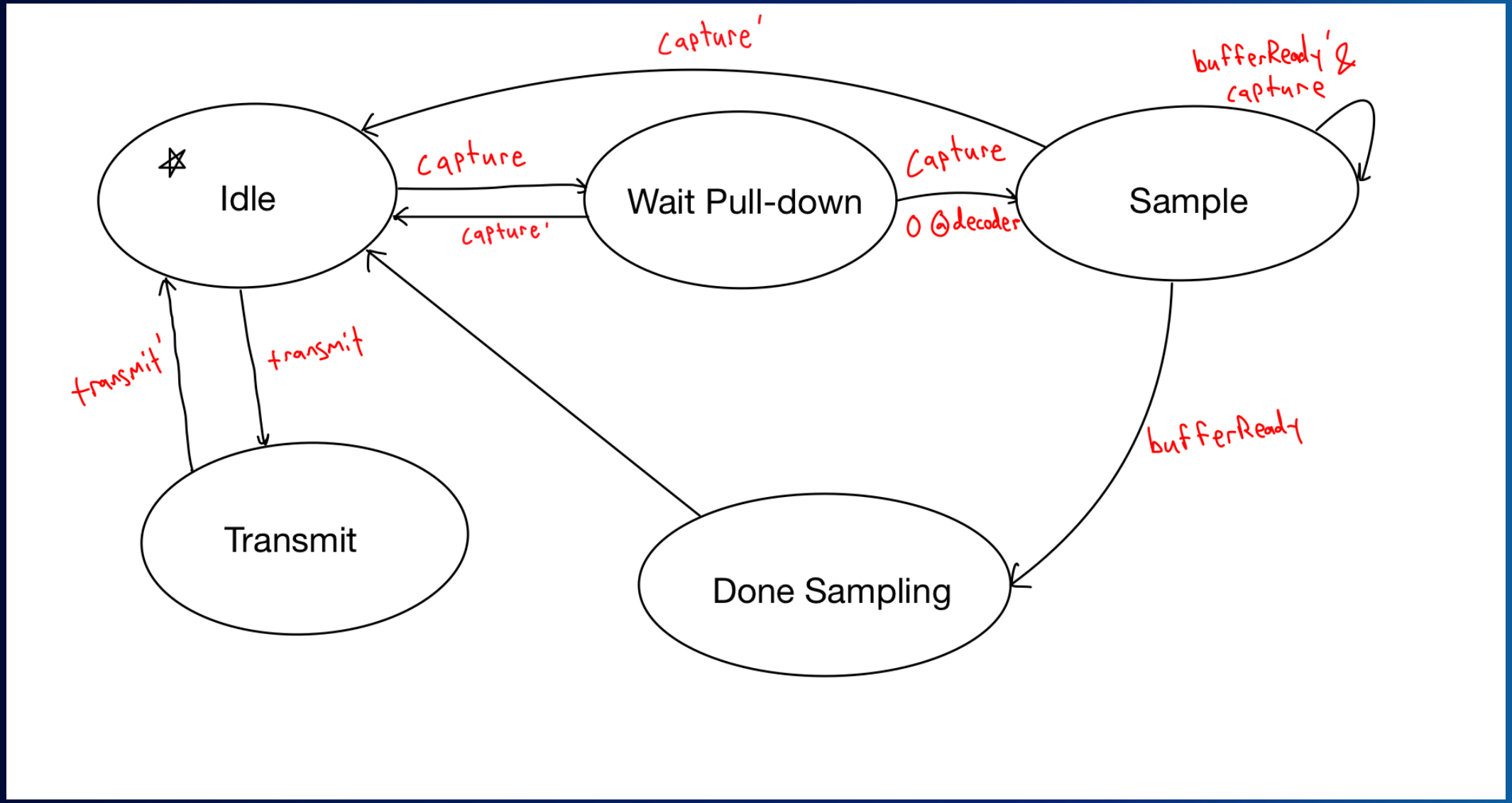
setSample(index, value)

```
//find the corresponding byte for the index
uint16_t byteIndex = index >> 3; //index >> 3equivalent to dividing by 8
//find the corresponding bit in the byte
uint8_t bitIndex = index & 7;    // index & 7 equivalent to modulo by 8
// we create a unique bit mask depending on the the bit index
if (value == 1)
{
    recieve_buffer[byteIndex] |= (1 << bitIndex); // set the bit
}
else
{
    recieve_buffer[byteIndex] &= ~(1 << bitIndex); // clear bit
}
```

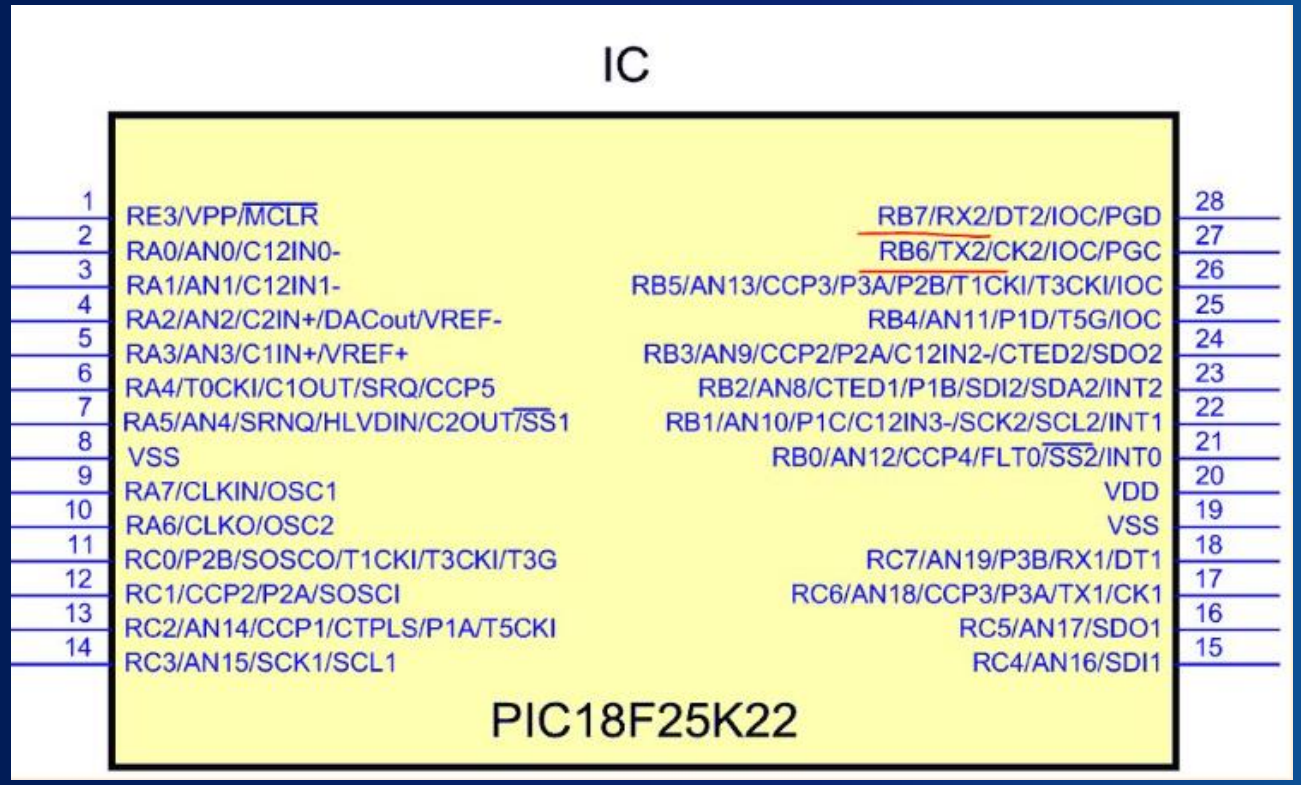
sendSample(buffer, currentByte, currentBit)

```
// load PWM based on value of current bit of current byte
// move the bit of interest to the least significant bit position
uint8_t bitValue = (recieveBuffer[currentByte] >> currentBit) & 0x01;
if (bitValue == 1)
{
    EPWM2_LoadDutyValue(IRLED_OFF); // turn LED OFF so DECODER GETS gets output 1
}
else
{
    EPWM2_LoadDutyValue(IRLED_ON); // turn LED ON SO DECODER GETS gets output 0
}
```

Finite State Machine



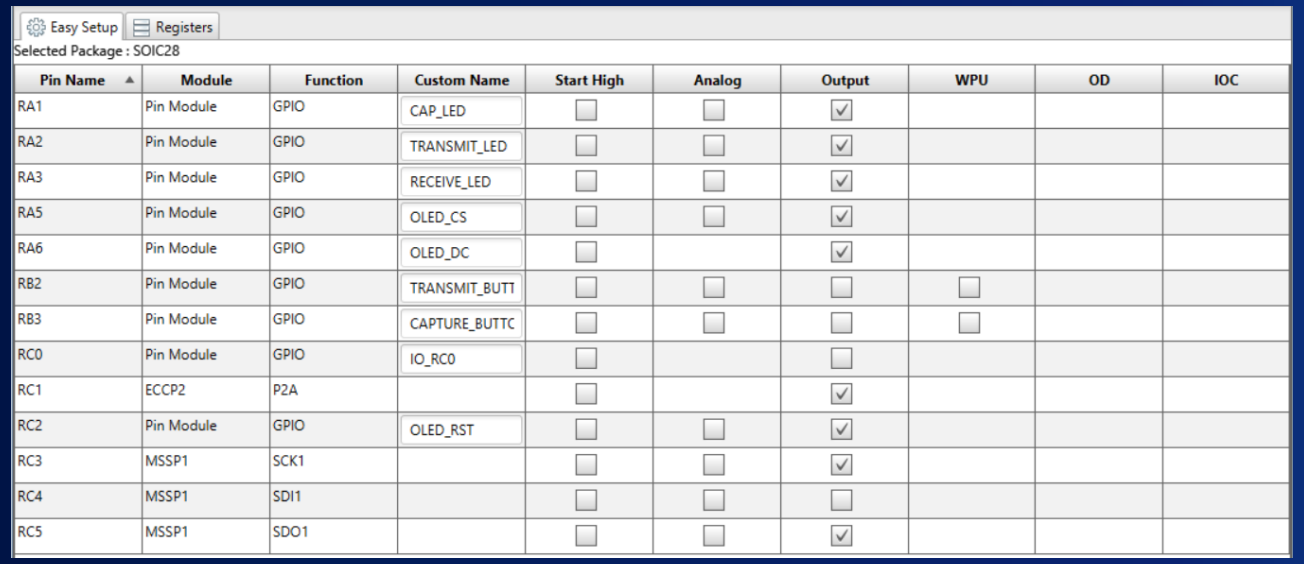
OLED Subsystem: Hardware Connection



OLED Subsystem: Software Connection

Header Files:
-oled.h
-bitmap.h
-demo.h
-font.h
-time_delay.h
-datatypes.h
-ide.h

Source Files:
-oled.c
-bitmap.c
-demo.c
-font.c
-time_delay.c

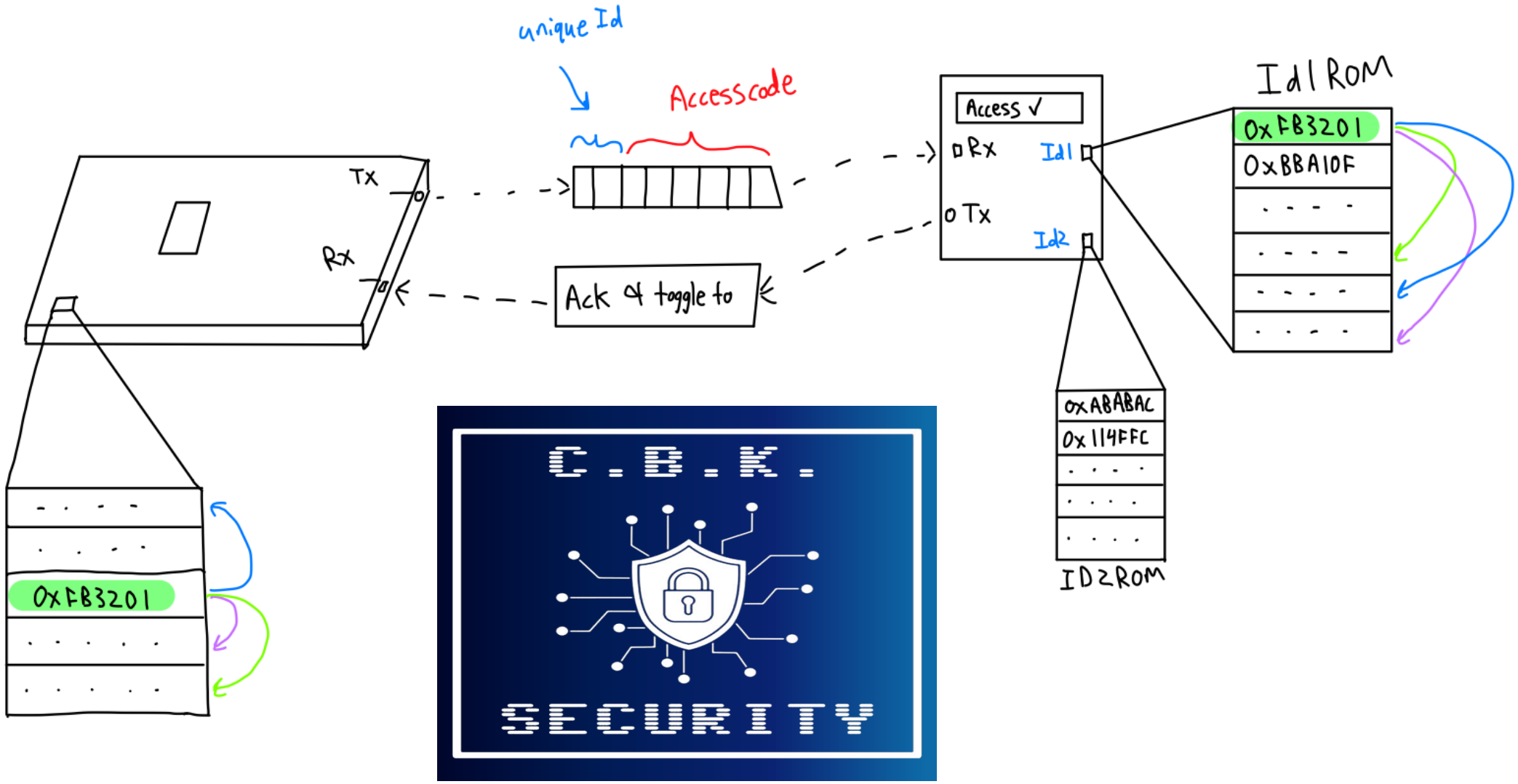


Easy Setup | Registers

Selected Package : SOIC28

Pin Name	Module	Function	Custom Name	Start High	Analog	Output	WPU	OD	IOC
RA1	Pin Module	GPIO	CAP_LED	☐	☐	☑			
RA2	Pin Module	GPIO	TRANSMIT_LED	☐	☐	☑			
RA3	Pin Module	GPIO	RECEIVE_LED	☐	☐	☑			
RA5	Pin Module	GPIO	OLED_CS	☐	☐	☑			
RA6	Pin Module	GPIO	OLED_DC	☐		☑			
RB2	Pin Module	GPIO	TRANSMIT_BUTT	☐	☐	☐	☐		
RB3	Pin Module	GPIO	CAPTURE_BUTTC	☐	☐	☐	☐		
RC0	Pin Module	GPIO	IO_RC0	☐		☐			
RC1	ECCP2	P2A		☐		☑			
RC2	Pin Module	GPIO	OLED_RST	☐	☐	☑			
RC3	MSSP1	SCK1		☐	☐	☑			
RC4	MSSP1	SDI1		☐	☐	☐			
RC5	MSSP1	SDO1		☐	☐	☑			

C.B.K Protocol


unique Id
Accesscode
Tx
Rx
Ack & toggle to
Access ✓
Rx
Tx
Id1
Id2
Id1 ROM
0xFB3201
0xBBA10F
0xABABAC
0x114FFC
ID2ROM
0xFB3201
C.B.K.
SECURITY

QUESTIONS