



Development of high-performance but low-cost sensing node for intelligent monitoring system

Department of Electrical and Computer Engineering, Undergraduate Student: **Yuzhen Chen**, Boning Cai
Mentor: Department of Civil Engineering, P.H.D Wentao Wang

Introduction

As society grows and urbanization continues to increase, cities are becoming larger and more diverse in their social functions, and at the same time, the number of potential hazards in cities has increased dramatically, such as whether the city's infrastructure is still robust after daily exposure to the elements, whether there are excessive pollutants in the river water, whether there are aging circuits and fire risks in the power lines and signal networks that travel through the city, and so on. In a modern city of this size, it is not practical to manually count every data and report the potential dangers. So, in this context, the research project becomes very important. We want to make reliable and stable sensors for detecting information about the external environment at low prices, there are many types of sensors and different sensors will be responsible for different functional boards. Thousands of sensors will be dropped by drones to all corners of the city, because of their tiny size, do not require continuous charging, and inexpensive, so after they lose their ability to work in the city can be naturally degraded without the need for manual recovery. These sensors will be able to weave a web of information in the city, reporting potential crises to detection centers in real time.

Project method

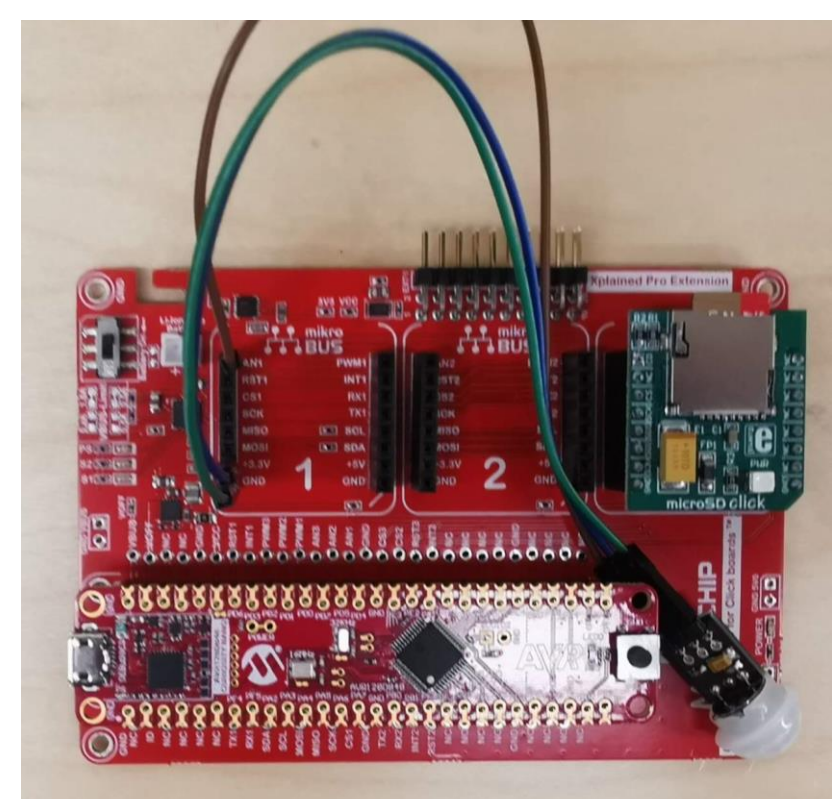
- Individuals targeted by the project are big cities that need sensors to collect specific data such as temperature, distance, and so on.
- Implement microcontroller AVR128DB48 onto the bread as well as SD card receiver and infrared sensors
- Connect microcontroller AVR128db48 with the computer.
- Open MPLAB to check connection with AVR128DB48
- Implement simple projects such as "Hello World" to check the operation of UART.
- Build connection for microcontroller in MPLAB.
- Build and implement the code from MPLAB to the microcontroller.
- Open Data Visualizer in MPLAB to capture reading from the sensor.
- Change to another water sensor and repeat the step above.



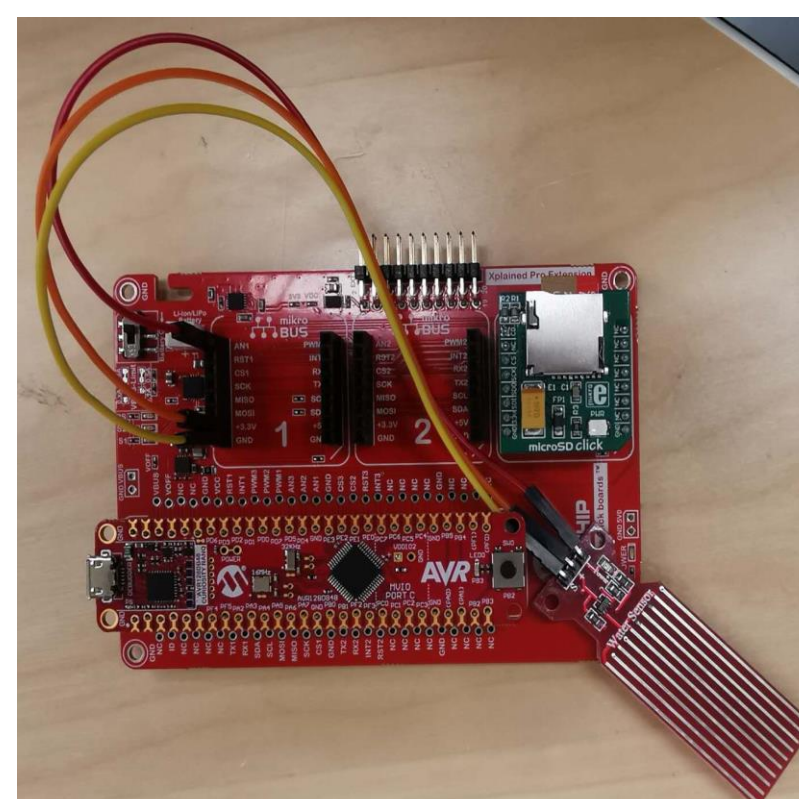
Infrared sensor



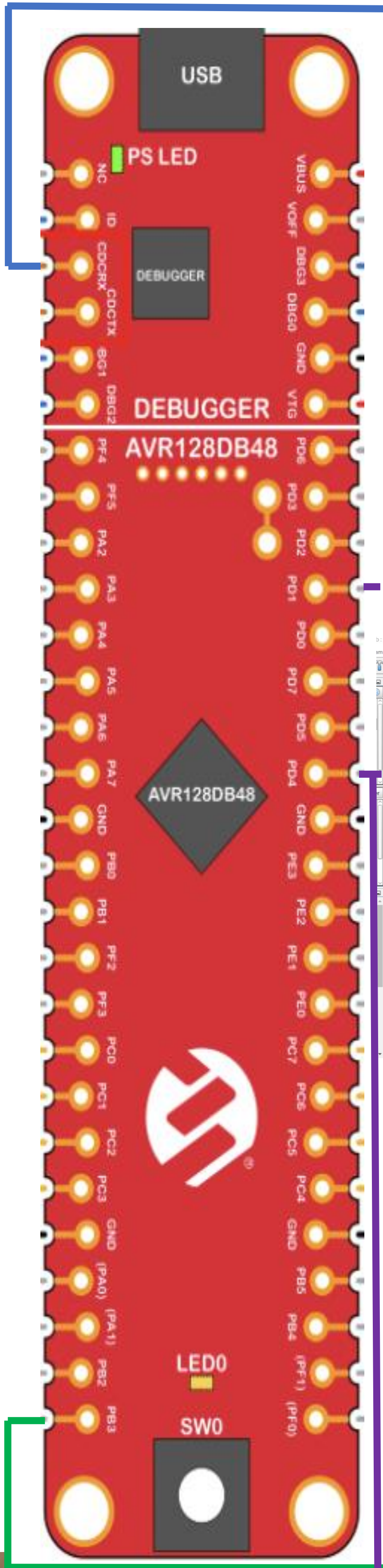
Water sensor



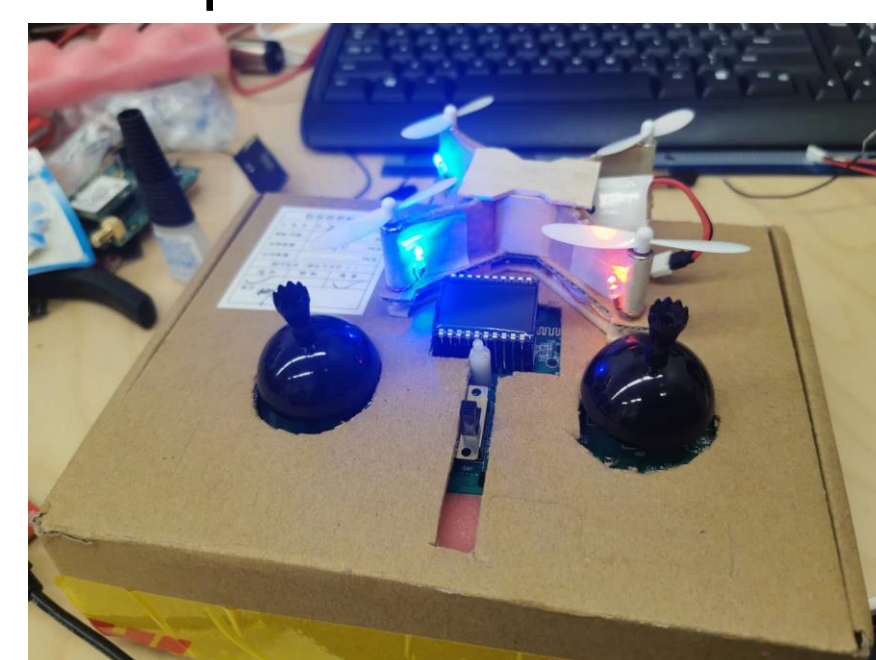
Infrared sensor connection



Water sensor connection



Displacement sensor



Drone and handle

Project Demo

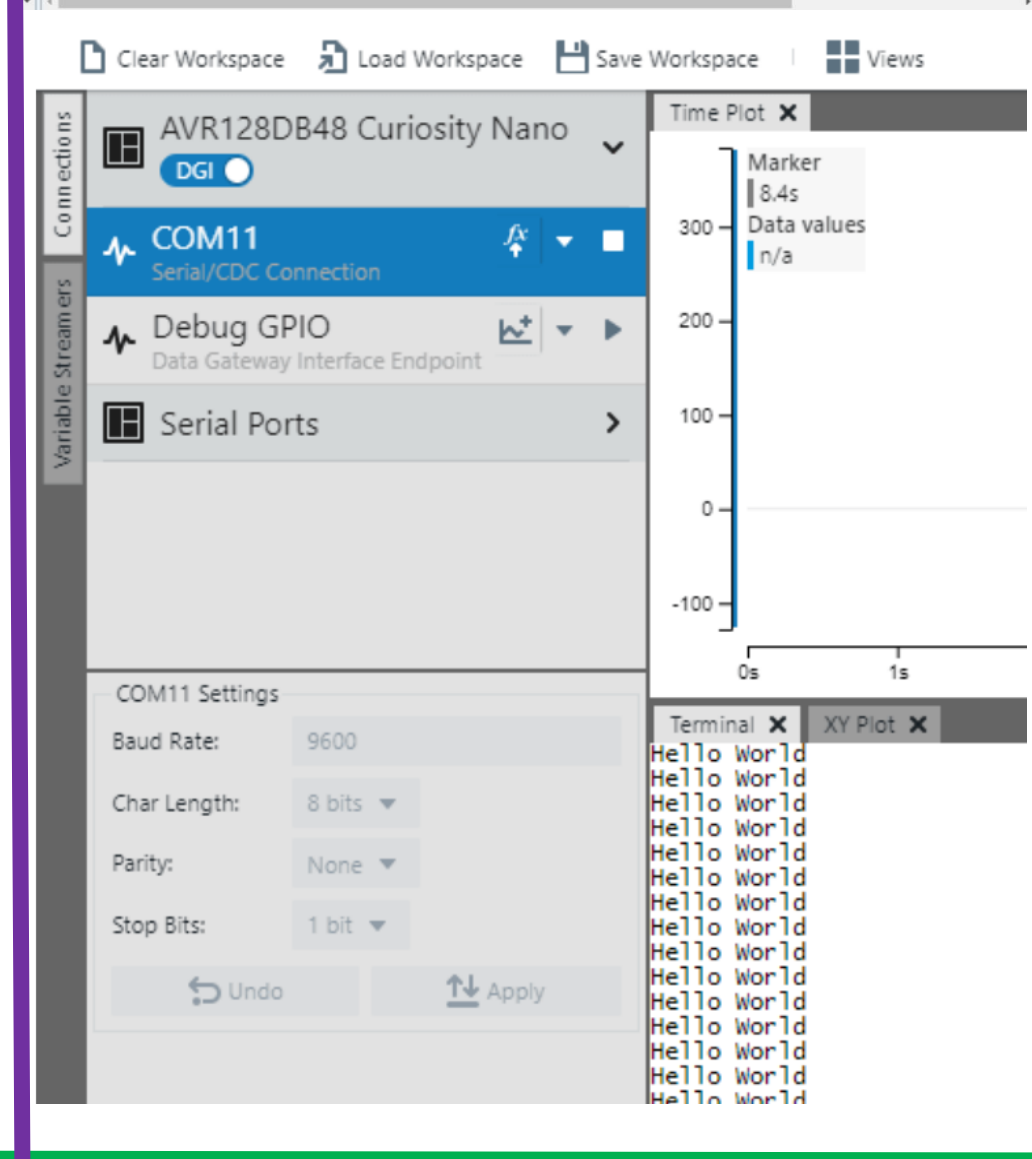
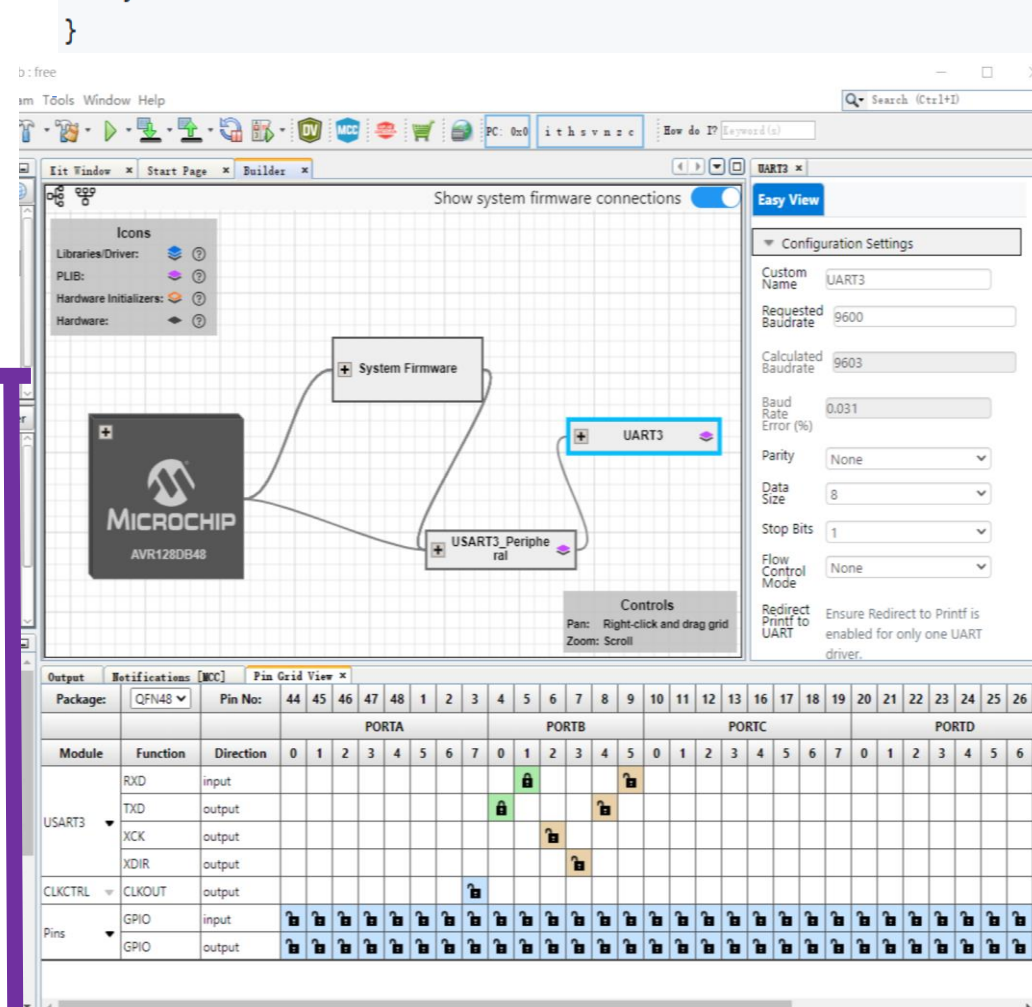
• Hello World print

```
#define STRING_END '\0'

int main(void)
{
    SYSTEM_Initialize();

    while(1)
    {
        /* Configured USART3 for printing, so this is one way of
        * writing text over USART.
        */
        printf("Hello ");

        /* Another way is to use the USART3_Write function directly
        * Can also be used to send uint8_t values directly */
        char world[] = "World \n";
        for(uint8_t i = 0; world[i] != STRING_END; i++){
            while(!USART3_IsTxReady());
            USART3_Write(world[i]);
        }
    }
}
```



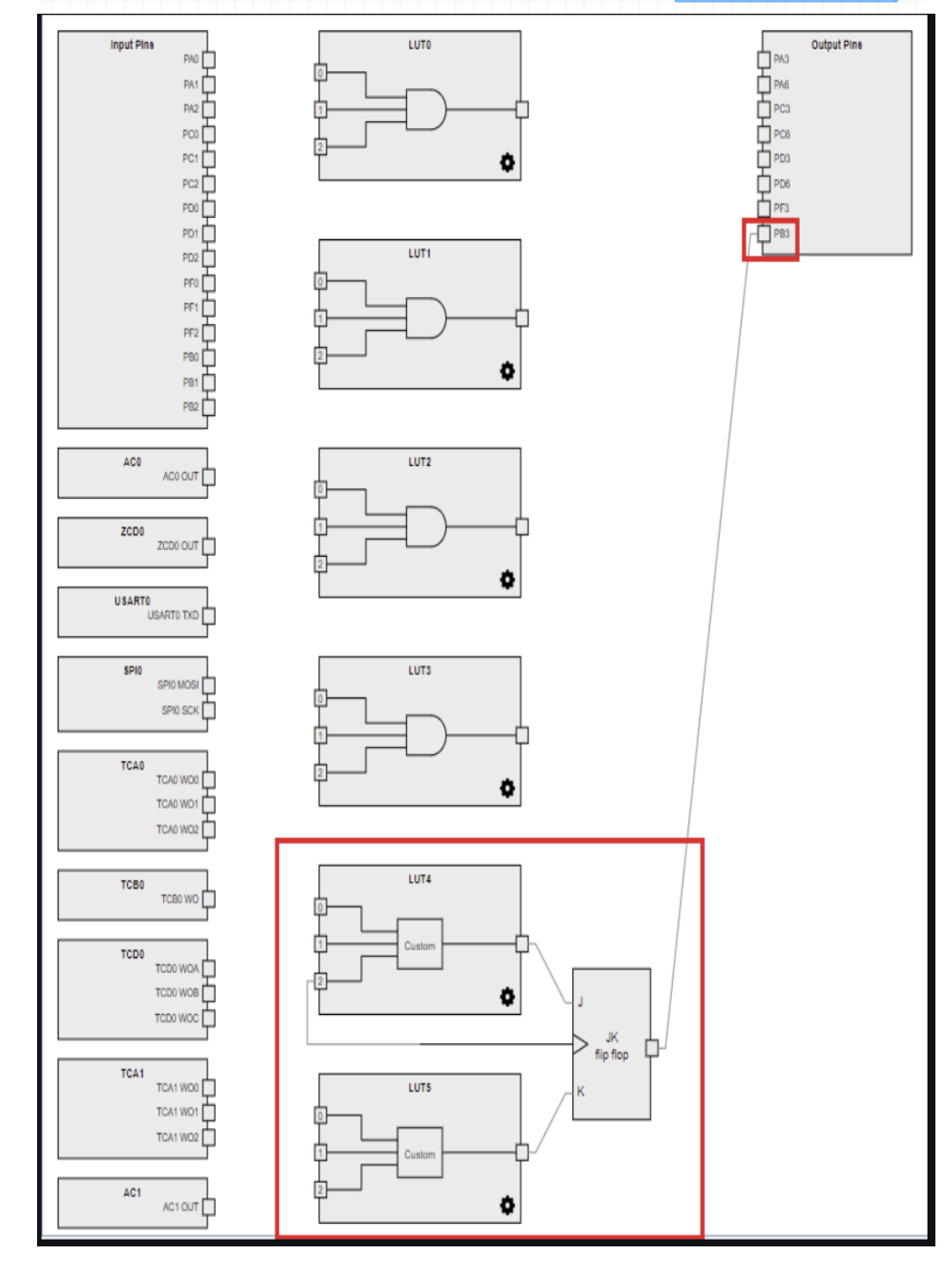
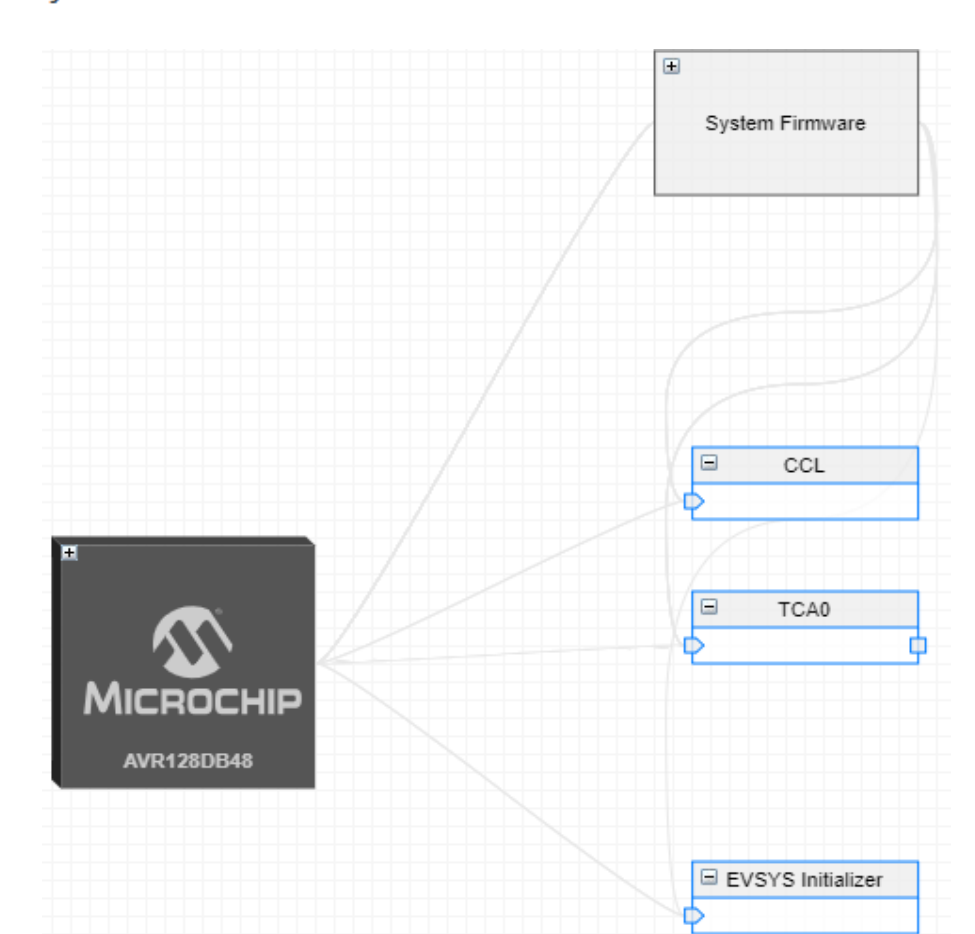
• LED Blinking

```
#include "mcc_generated_files/system/system.h"

/*
 * Main application
 */

int main(void)
{
    SYSTEM_Initialize();

    while(1)
    {
    }
}
```



• Analog Read Serial Write

```
/* Add constants for the data stream sent out from USART3: */

#define START_TOKEN 0x00 /* Start Frame Token */
#define ADC_RESULT_SIZE 16 /* Number of bits of measurement chosen for ADC */
#define ADC_RESOLUTION 4 /* Number of resolutions chosen for ADC */

/* Make a function that sends one byte of data out on USART3: */
void USART3_SendByte(const uint8_t data)
{
    while(!USART3_IsTxReady()); /* Wait until USART3 Data Register empty */
    USART3_Write(data); /* Send byte */
}

/* Make a function that transmits the data stream on USART3 (start token, 16 bit measurement, end token): */
void USART3_SendDataStream(const uint8_t data)
{
    USART3_SendByte(START_TOKEN); /* Send start token */
    USART3_SendByte(data); /* Send first 8 bits of measurement (2nd byte) */
    USART3_SendByte(data); /* Send last 8 bits of measurement (3rd byte) */
    USART3_SendByte(END_TOKEN); /* Send stop token */
}

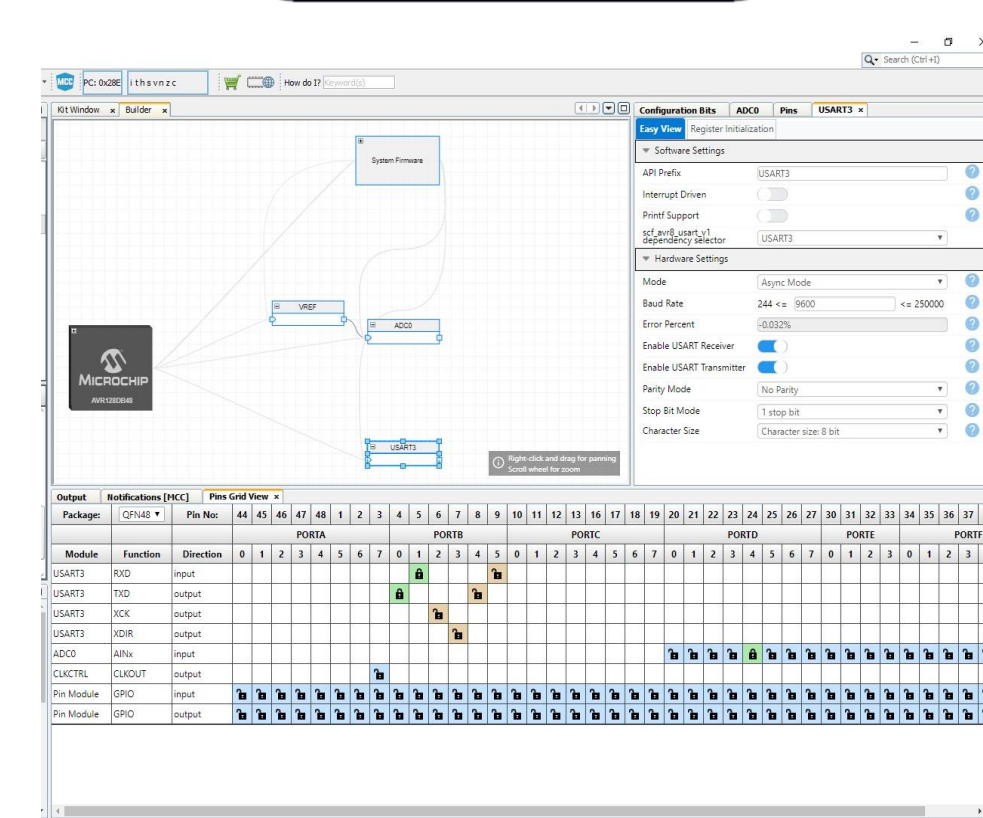
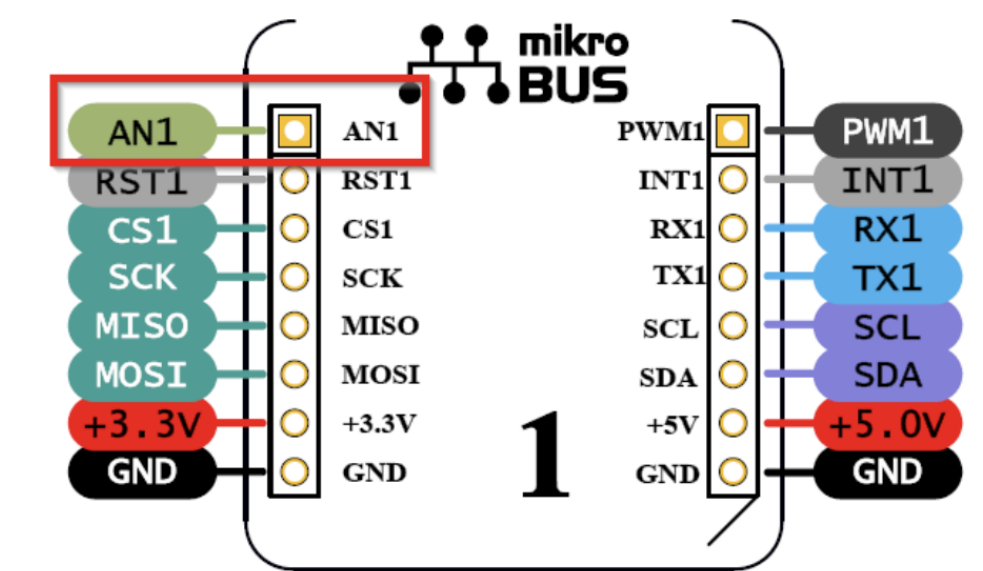
int main(void)
{
    SYSTEM_Initialize();

    while(1)
    {
        /* When variable is used, it will update value with latest ADC result */
        uint8_t measurement = ADC_Conversion(ADC_RESPOS_AIN1); /* Divide result by number of resolutions */
        USART3_SendDataStream(measurement); /* Send data stream out on USART3 */
    }
}
```

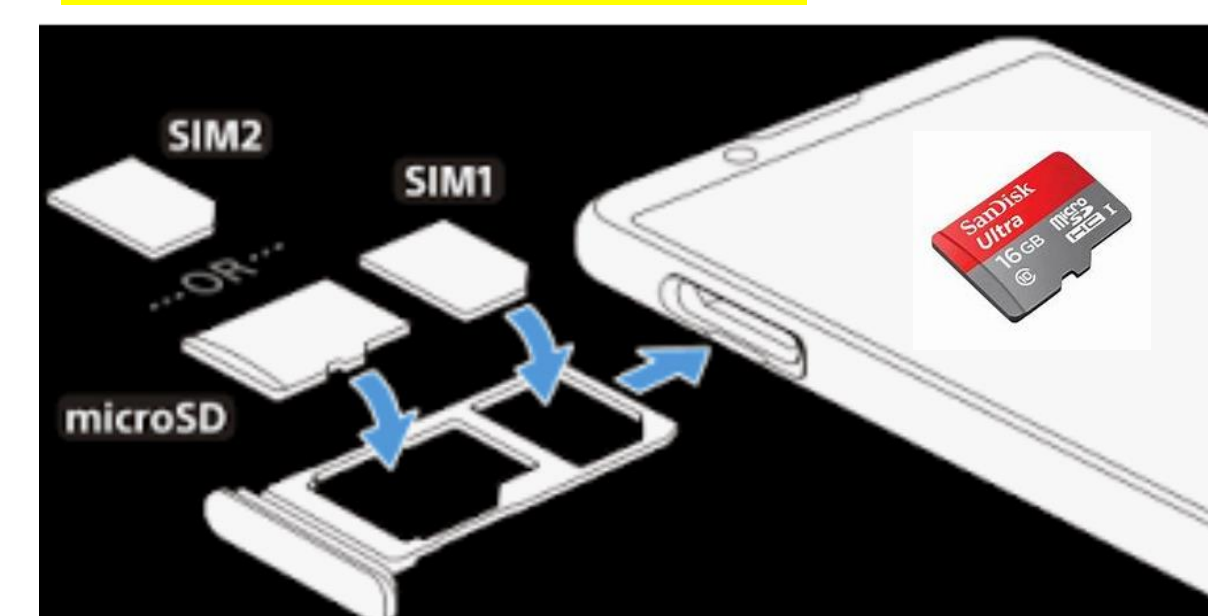
PINOUT DIAGRAM

This table shows how the pinout on POT click 3 corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin	1	AN	PWM	16	NC
Regulated Voltage Out	VO	1	AN	PWM	16	NC
	NC	2	RST	INT	15	NC
Chip Select	CS	3	CS	RX	14	NC
Serial Clock	SCK	4	SCK	TX	13	NC
Serial Data Out	SDO	5	MISO	SCL	12	NC
	NC	6	MOSI	SDA	11	NC
Power Supply	3.3V	7	3.3V	5V	10	5V
Ground	GND	8	GND	GND	9	GND



Future Plan



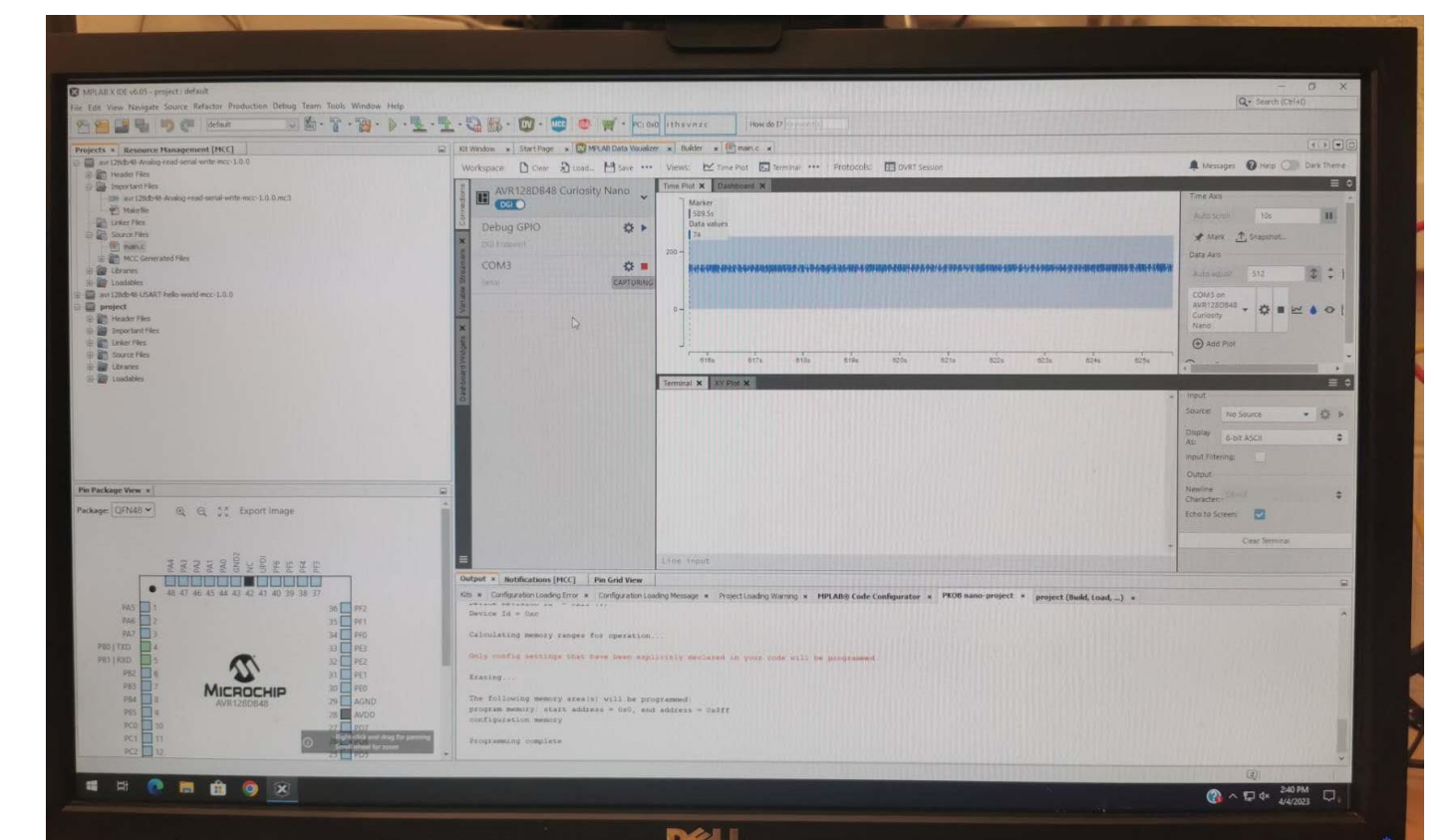
Use SD card to transfer sensor data



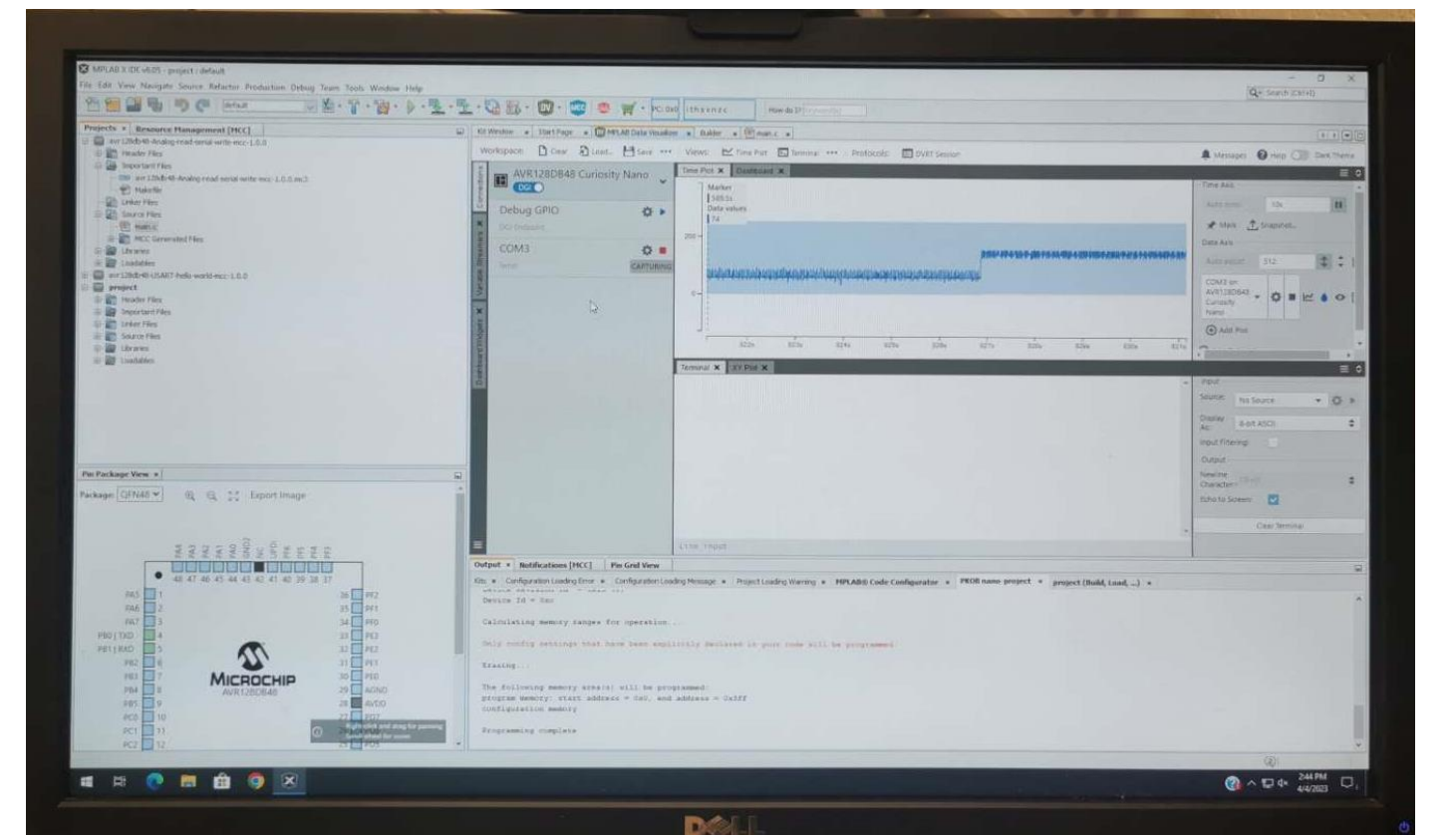
Use Drones to deliver our sensors

Project Result

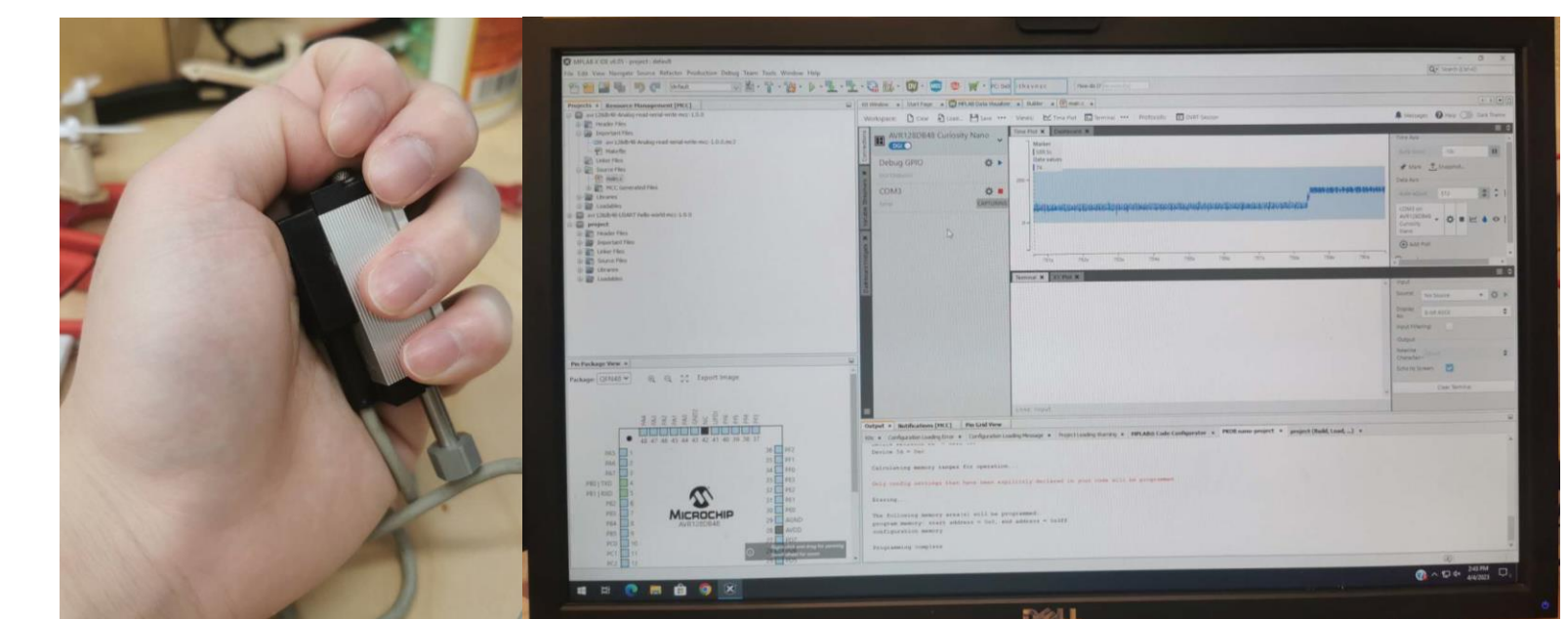
The result of this project is after several demos in MPLAB, we can successfully implement our code into our sensors. After implementing our water, infrared and displacement sensor, we can successfully collect data. Thus, our sensor can be used in future real application such as water, displacement, and number of people sensing.



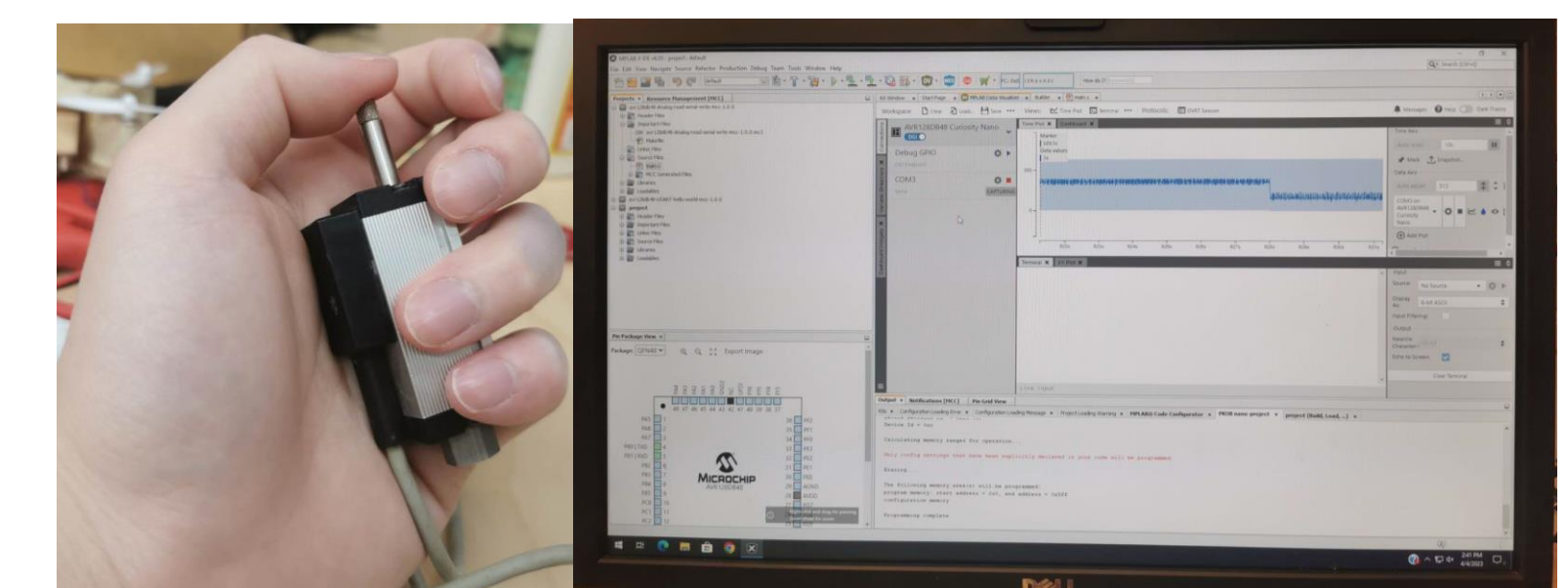
Infrared sensor reading when people passing



Water sensor reading when detect water

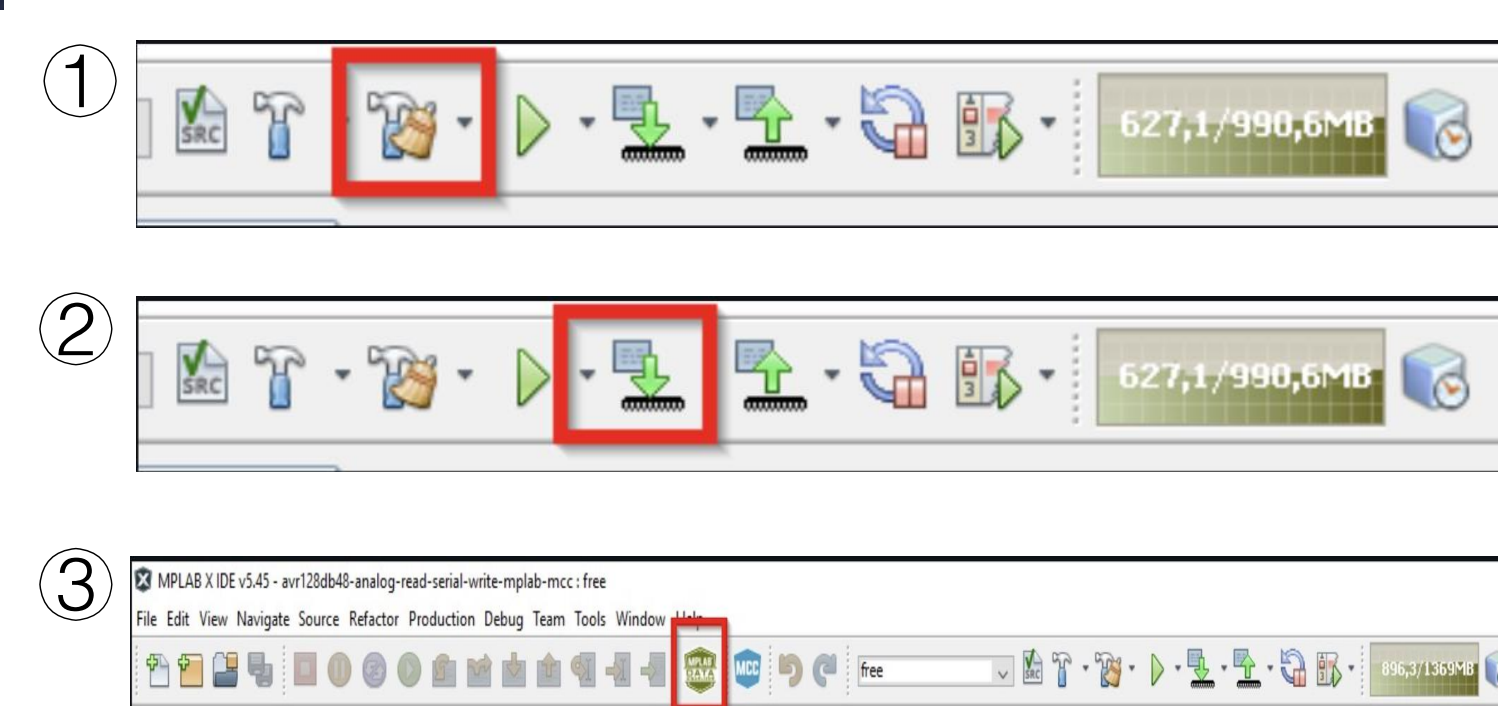


Displacement sensor reading when pushing



Displacement sensor reading when releasing

• How to start running



Reference

<https://github.com/microchip-pic-avr-examples/avr128db48-hello-world-over-uart-mplab-mcc>
<https://github.com/microchip-pic-avr-examples/avr128db48-blink-led-ccl-mplab-mcc>
<https://github.com/microchip-pic-avr-examples/avr128db48-analog-read-serial-write-mplab-mcc>