ECE 372A

#WeDidTheThing

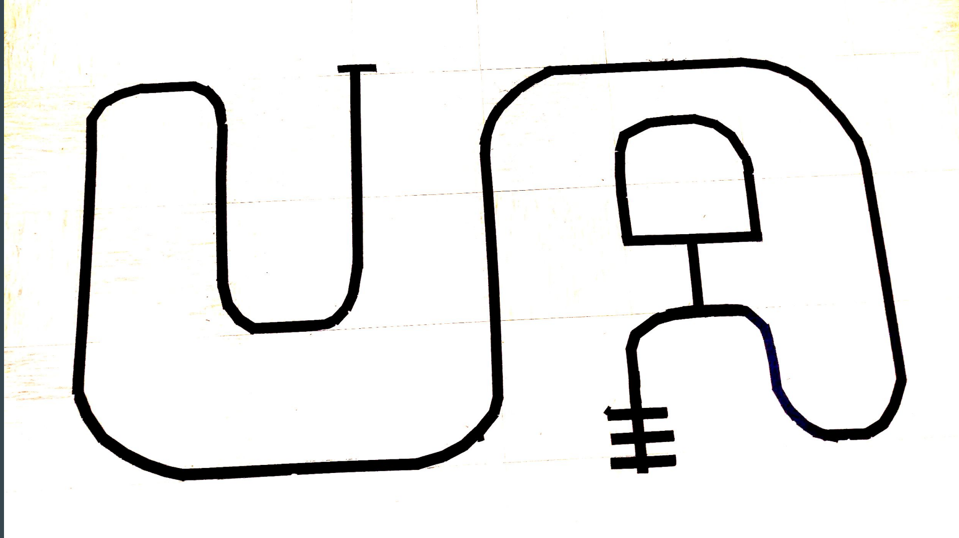
Matt Dzurick
Brett Bushnell
Sydney Clark
Ryan Trumpinski

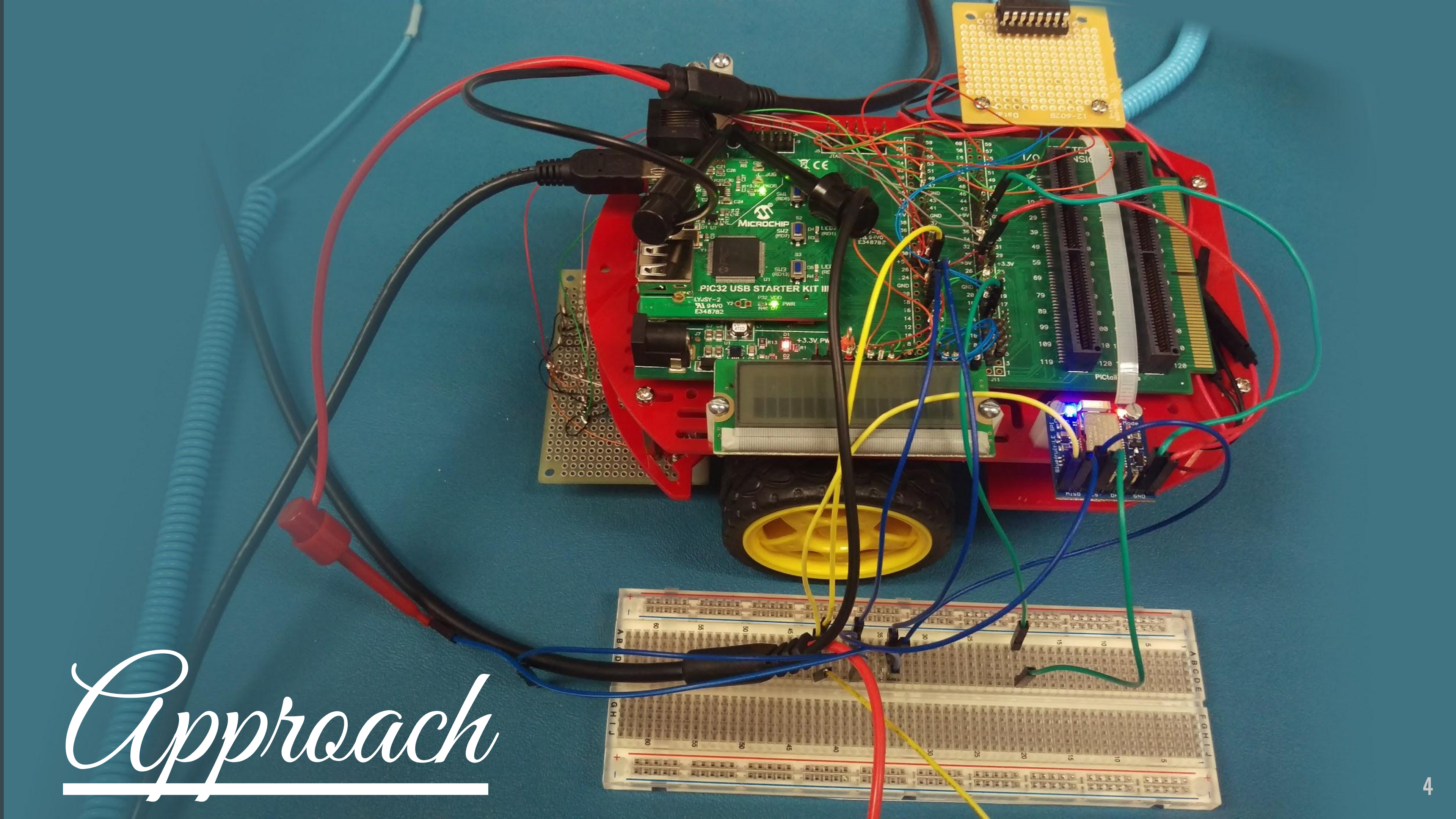
Outline

- ➢ Requirements
- ➢ Approach
- ➢ Hardware
- ➢ Software
- ➢ Deliverables
- ➢ Conclusion

Requirements

- Part 1
  - Follow the black tape line to the end of the track and back
- Part 2
  - Integrate Bluetooth sensor to communicate with Pic
- Part 3
  - Win the competition

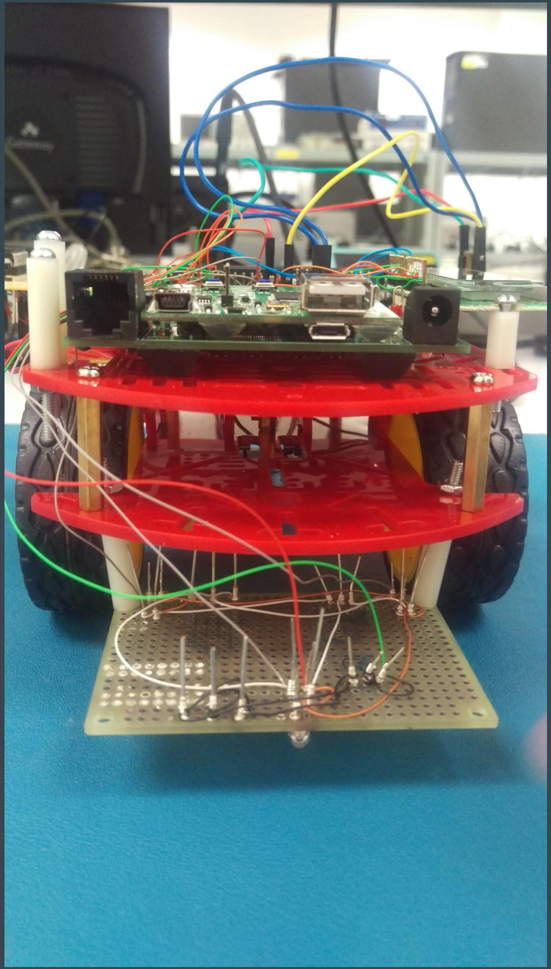




Approach

Divide and Conquer

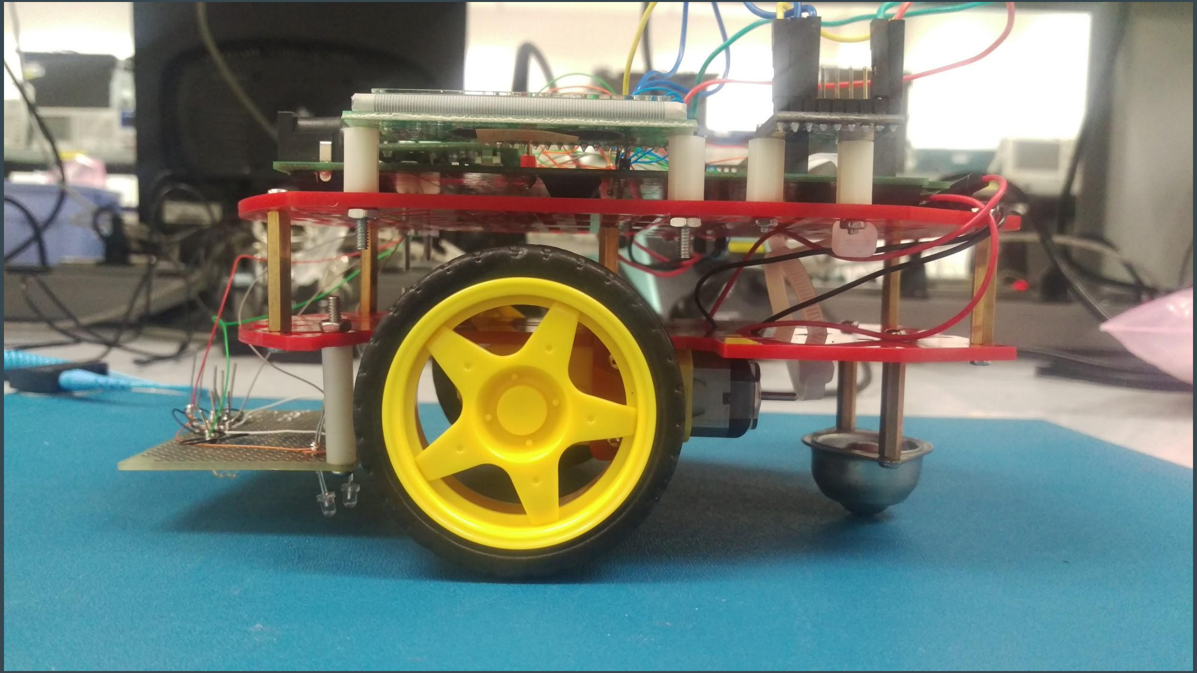
- ➢ Two members take hardware
- ➢ Two members take software



#WeDidTheThing

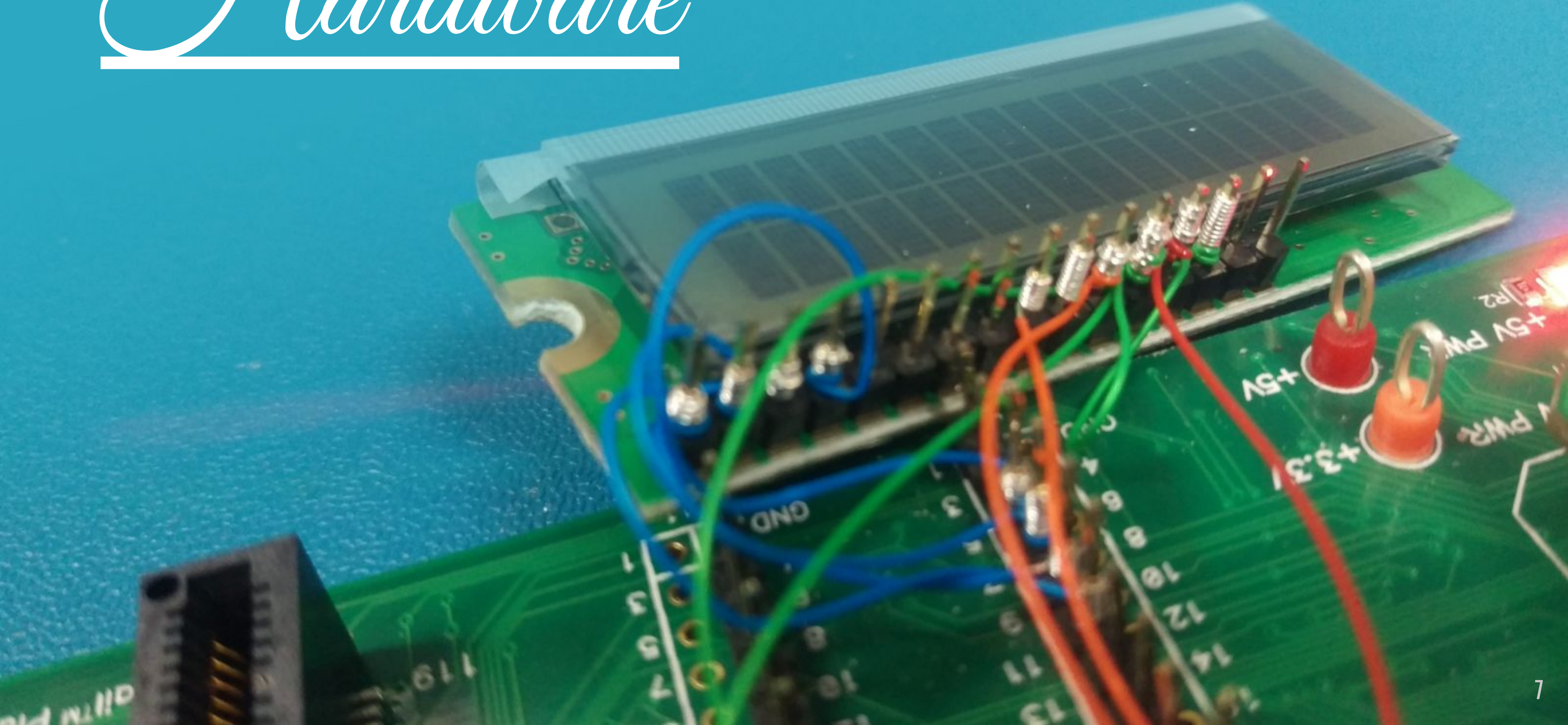
Result?

➢ Our robot is the first, the fastest, and the best!

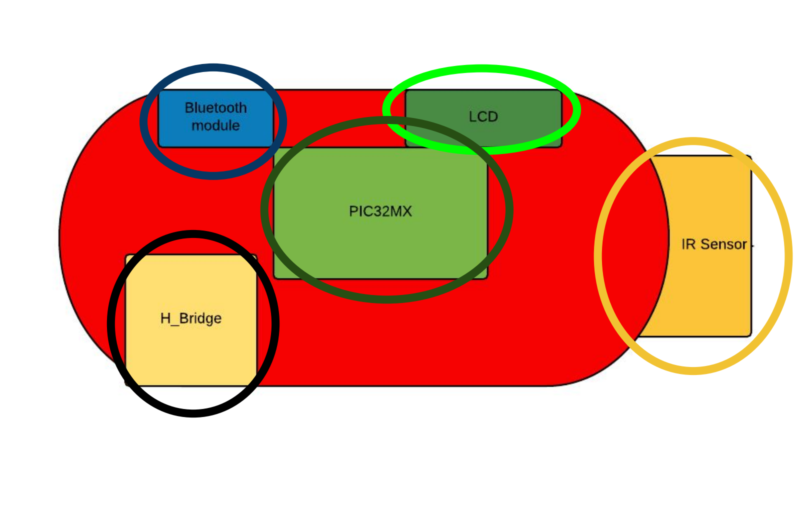


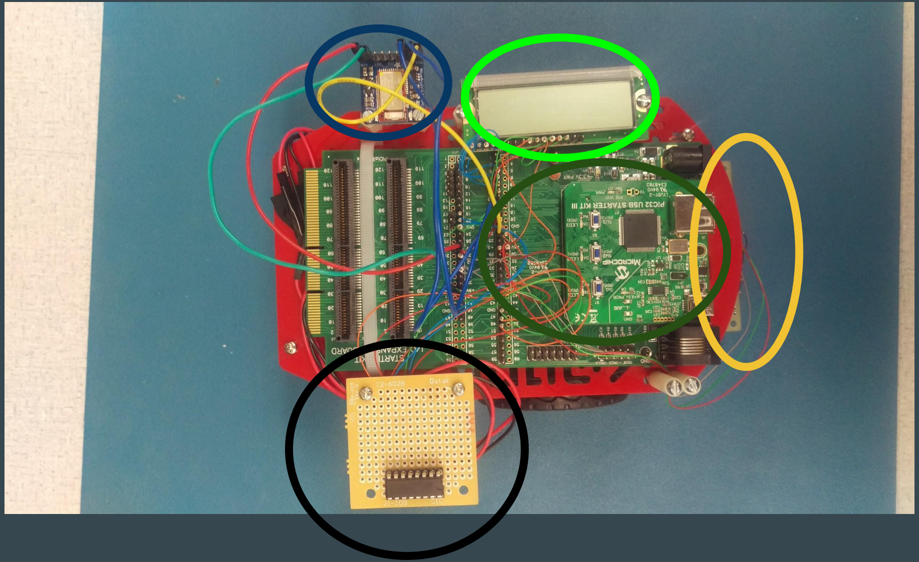
#WeDidTheThing

Hardware



Block Diagram



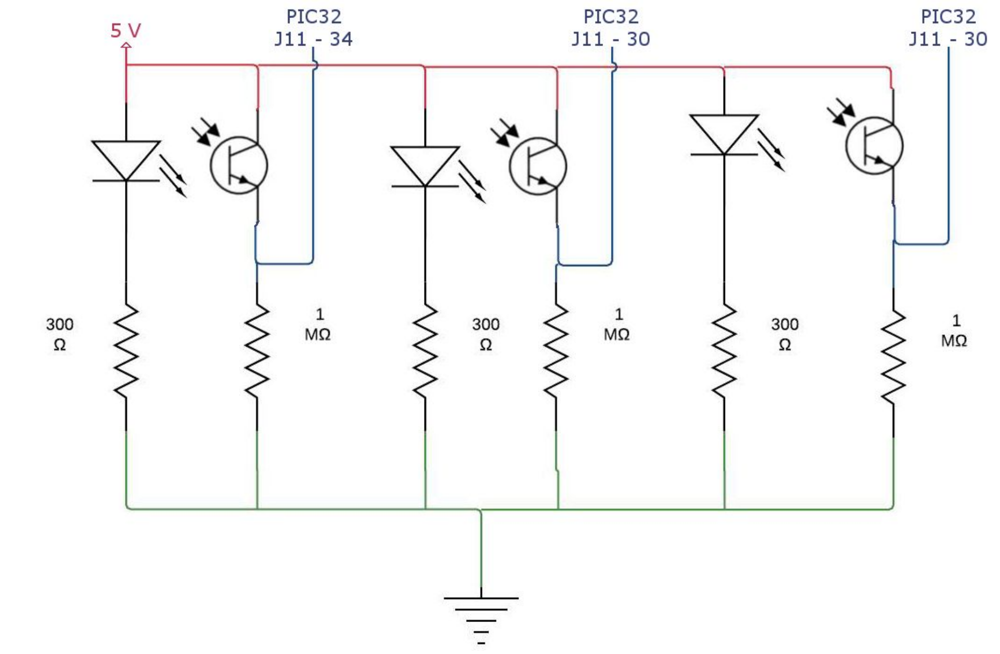


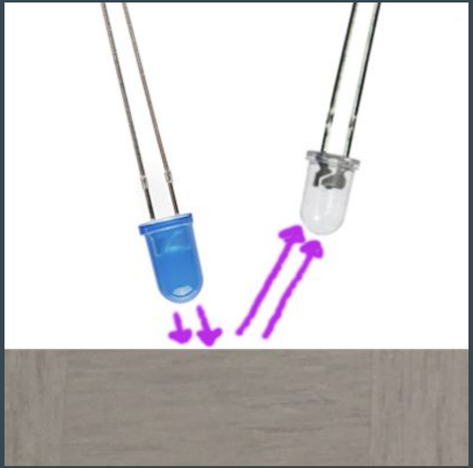
#WeDidTheThing

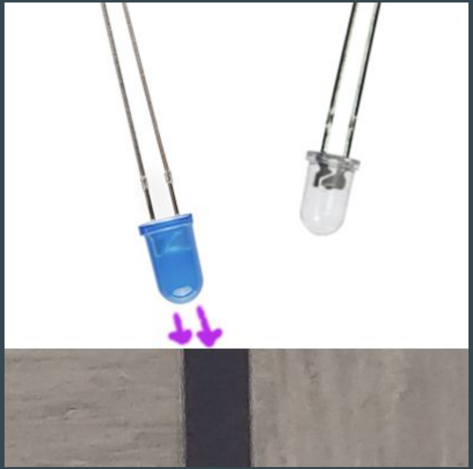
IR Sensors

- Three sets of LED and phototransistor.
- Arranged in a triangle formation.
- LED is closer to the ground than the phototransistor.

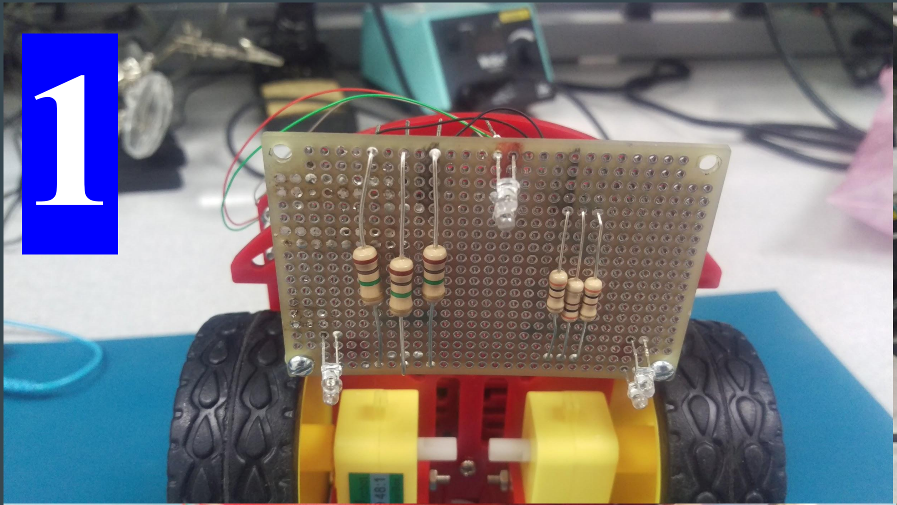
Circuit & Theory

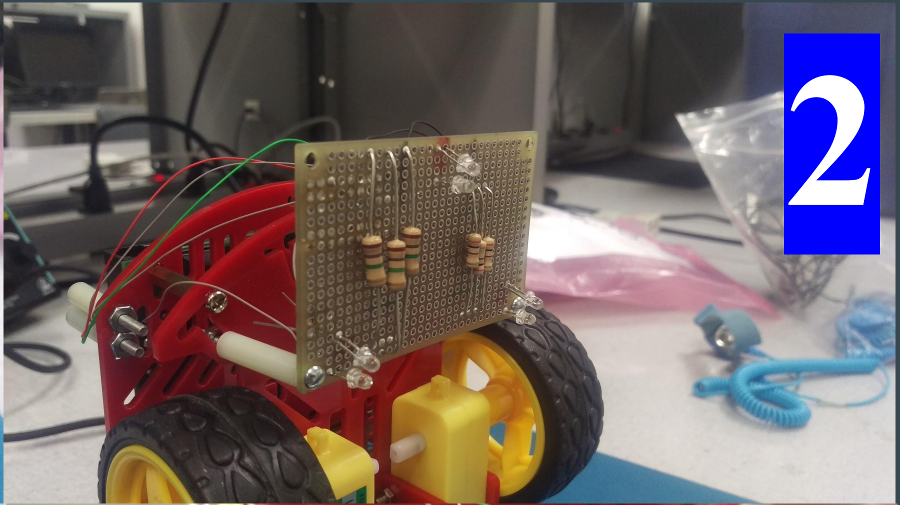


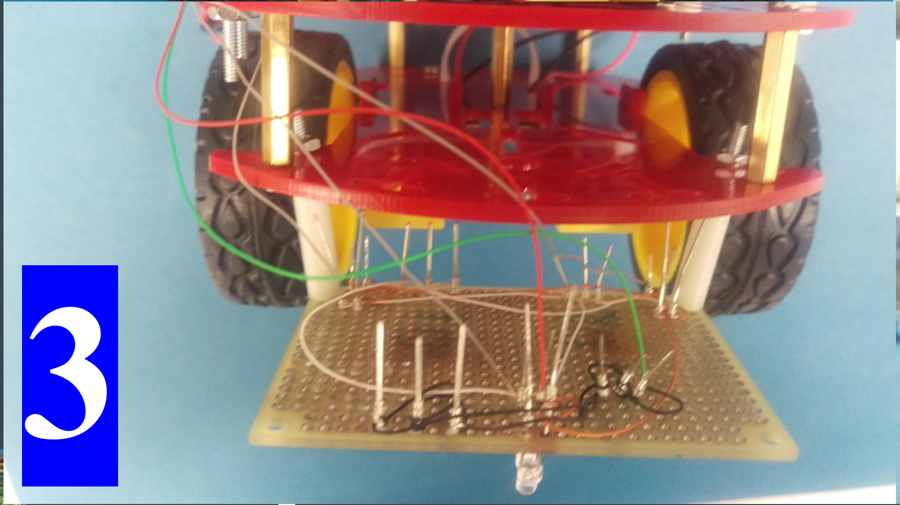


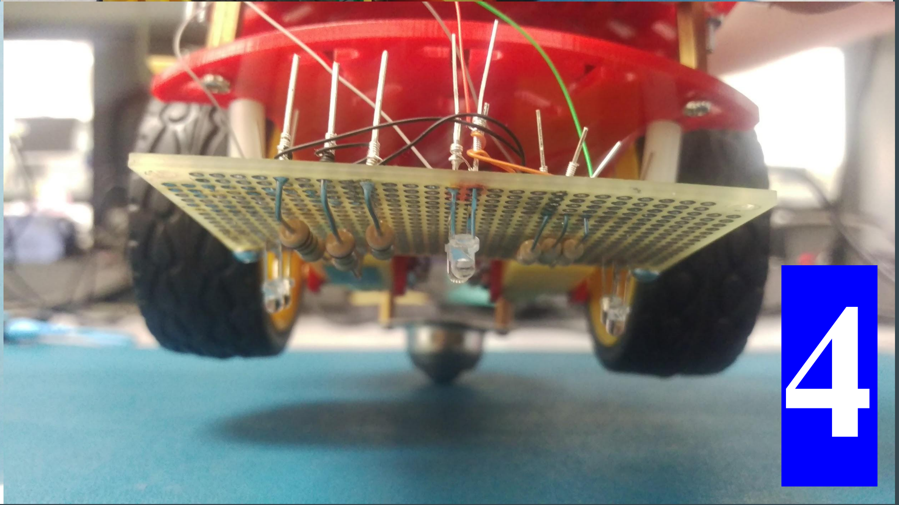


#WeDidTheThing

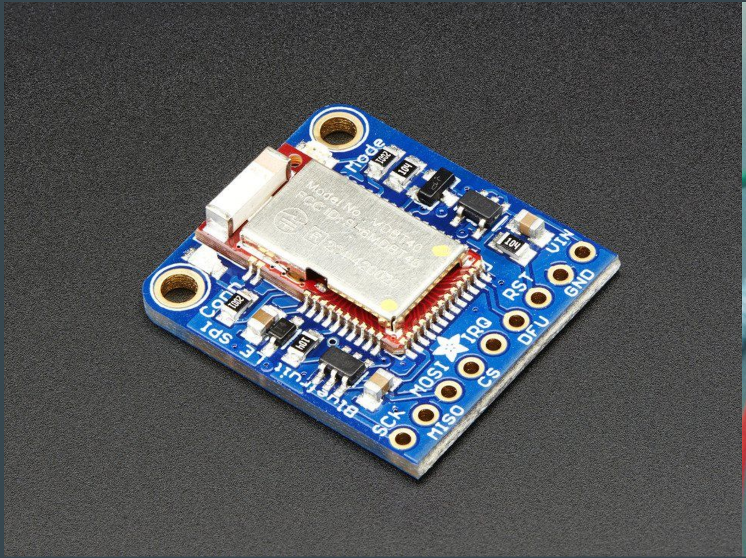

1

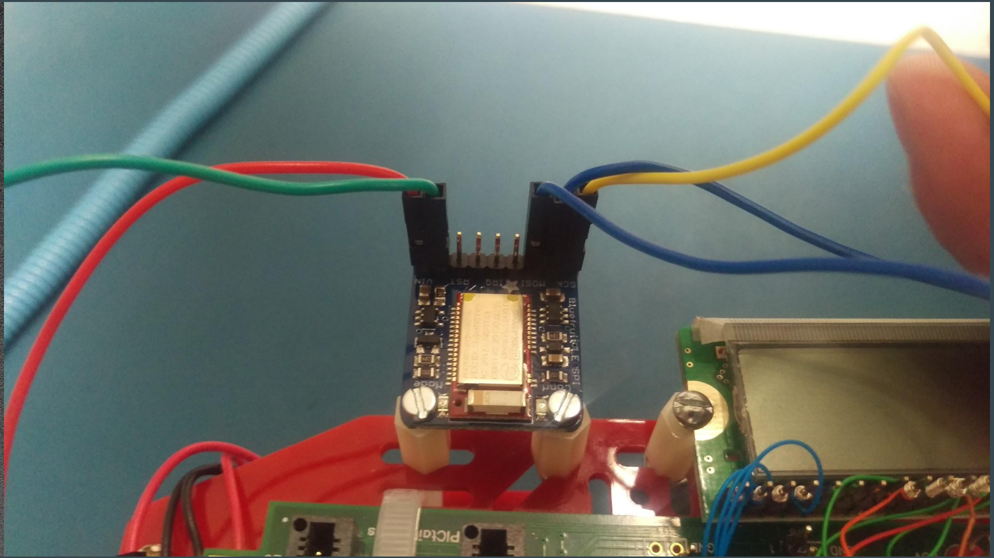

2


3

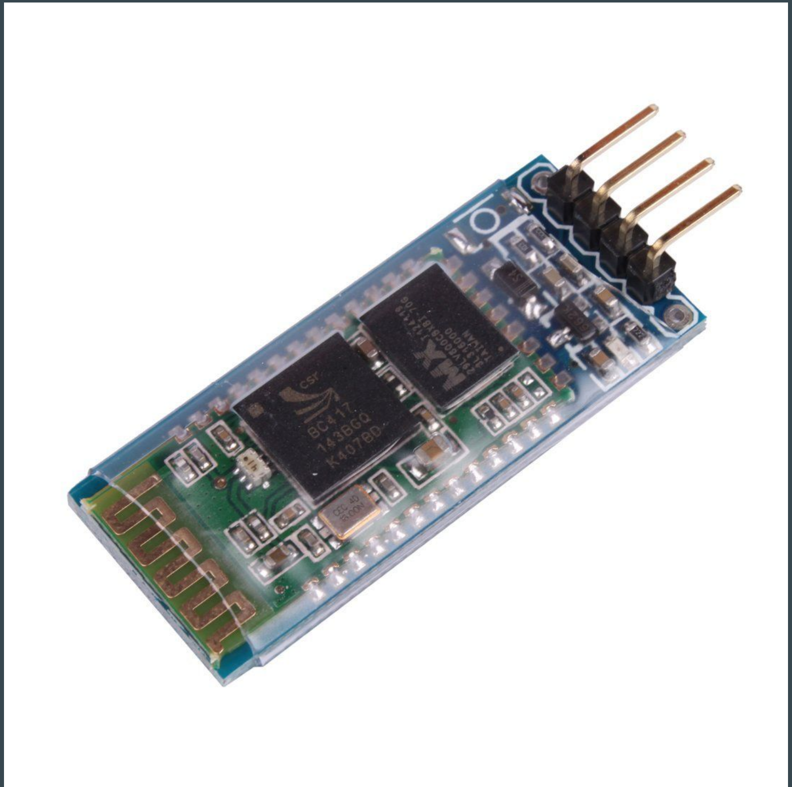

4

Adafruit Bluefruit LE SPI Friend-Bluetooth Low Energy



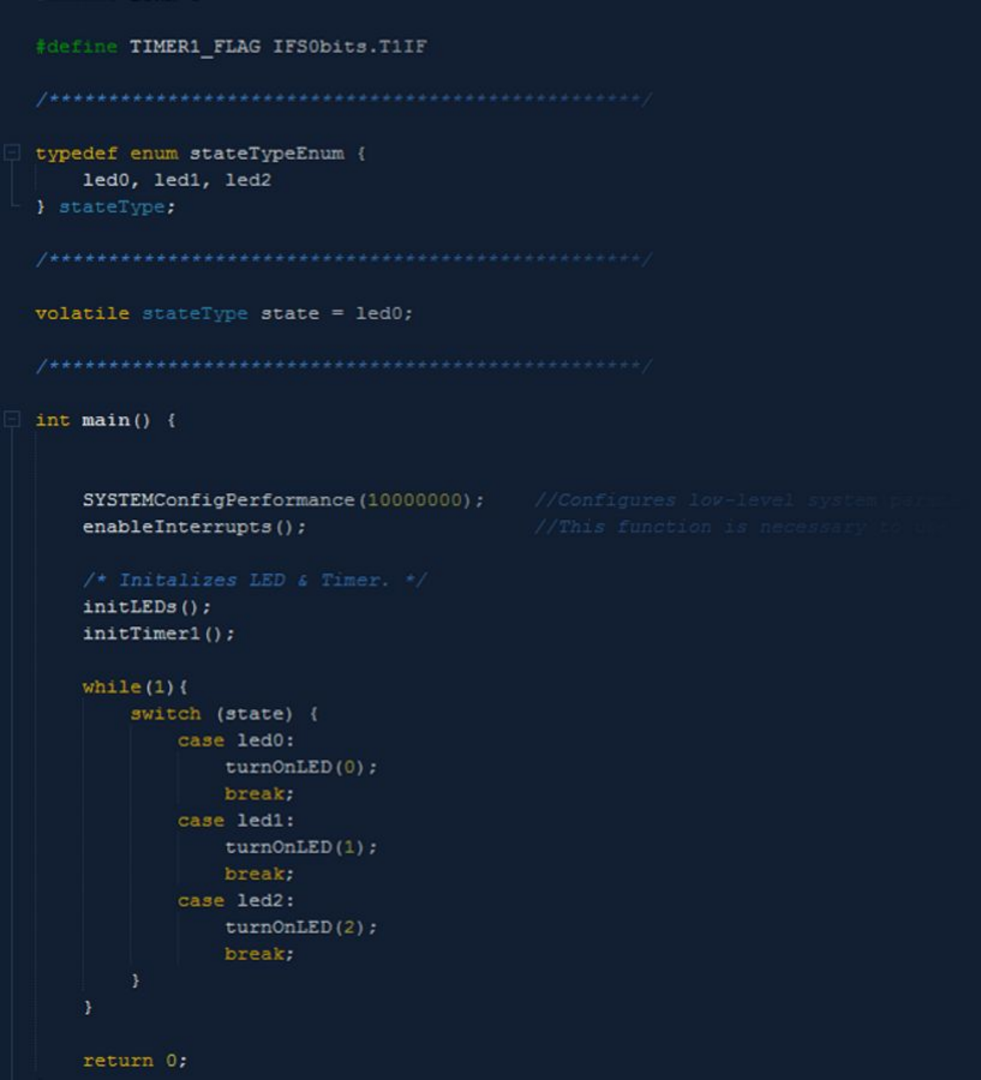


JBteck HC-06 Bluetooth



JBteck HC-06 Bluetooth

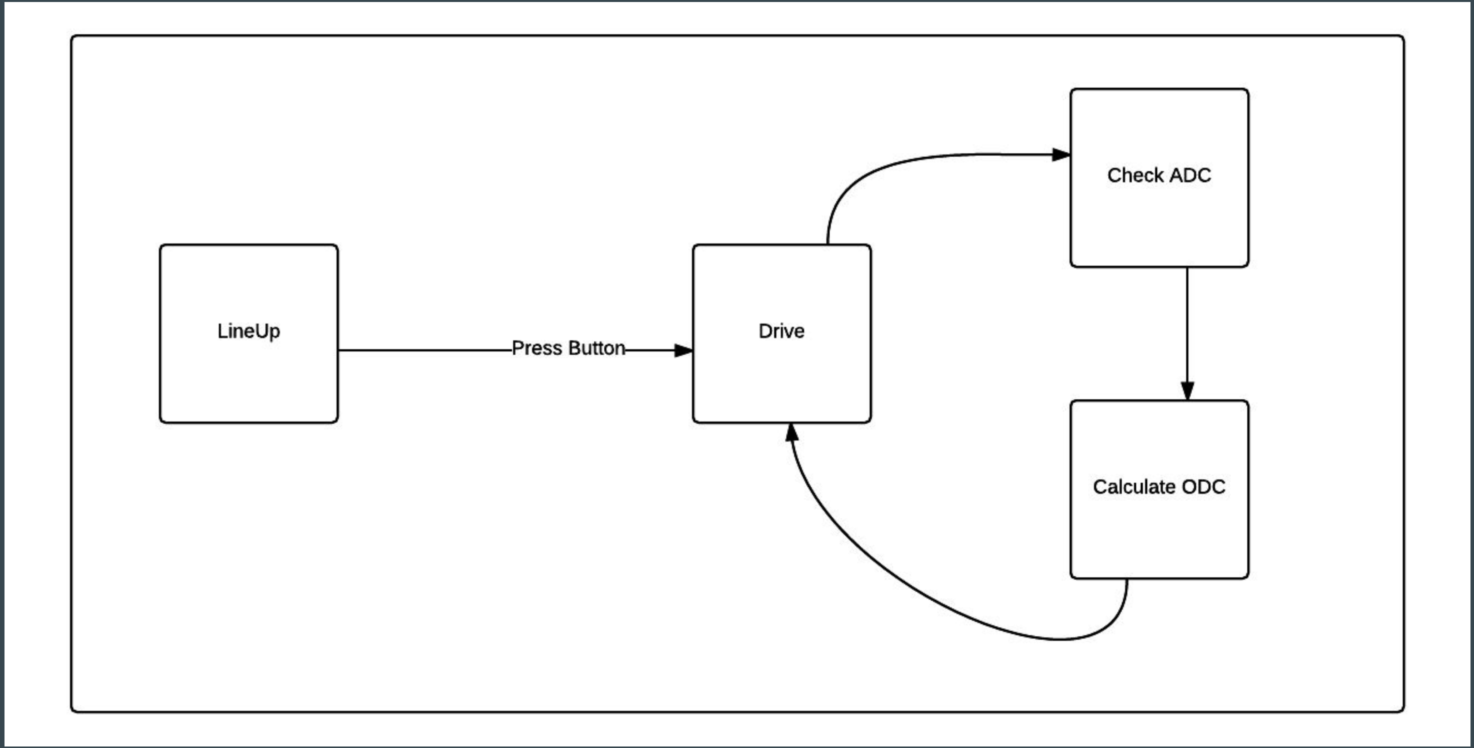
- Option between UART and SPI
  - SPI = SYNCHRONOUS
  - UART = ASYNCHRONOUS
- Using UART
  - Easier to implement on PIC32
  - Example code available from past classes



```
#define TIMER1_FLAG IFS0bits.T1IF

/************************************************/

typedef enum stateTypeEnum {
    led0, led1, led2
} stateType;

/************************************************/

volatile stateType state = led0;

/************************************************/

int main() {


    SYSTEMConfigPerformance(10000000);    //Configures low-level system para
    enableInterrupts();                   //This function is necessary to u

    /* Initalizes LED & Timer. */
    initLEDs();
    initTimer1();

    while(1){
        switch (state) {
            case led0:
                turnOnLED(0);
                break;
            case led1:
                turnOnLED(1);
                break;
            case led2:
                turnOnLED(2);
                break;
        }
    }

    return 0;
```

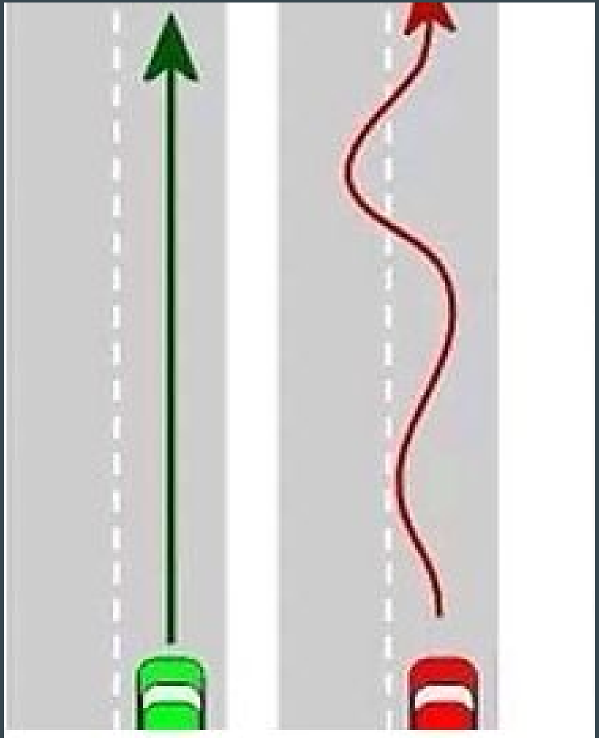
Software

Finite State Machine



Part 1

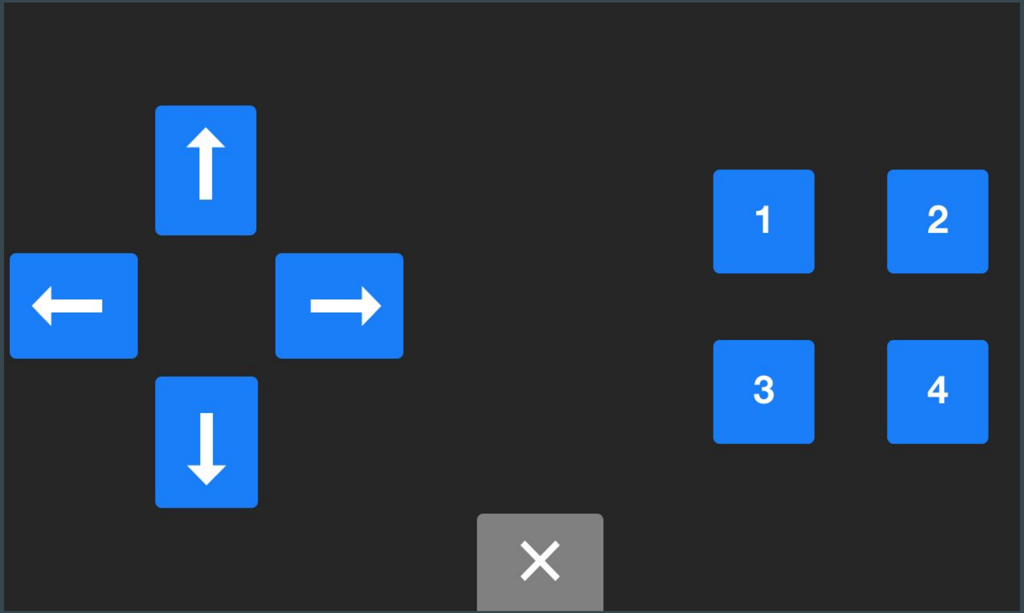
- LineUp state:
  - LED’s will light up when Robot is in position to start
- Drive:
  - Begin moving forward (Never look back)
- Check ADC
  - Check values read from the phototransistors
- Calculate ODC
  - Adjust speed of the motors



Expectation Reality

Part 2

- ➢ Integrate Bluetooth chip with PIC32
- ➢ Uses UART
  - ○ Baud rate: 9600
- ➢ Commands sent from control interface



Deliverables & Conclusion

Deliverables

- Code is available on Github
- Videos are available
- Final report is available

Conclusion

- Hardware is simple
- Software is simple
- Result: It works!

That's all Folks!