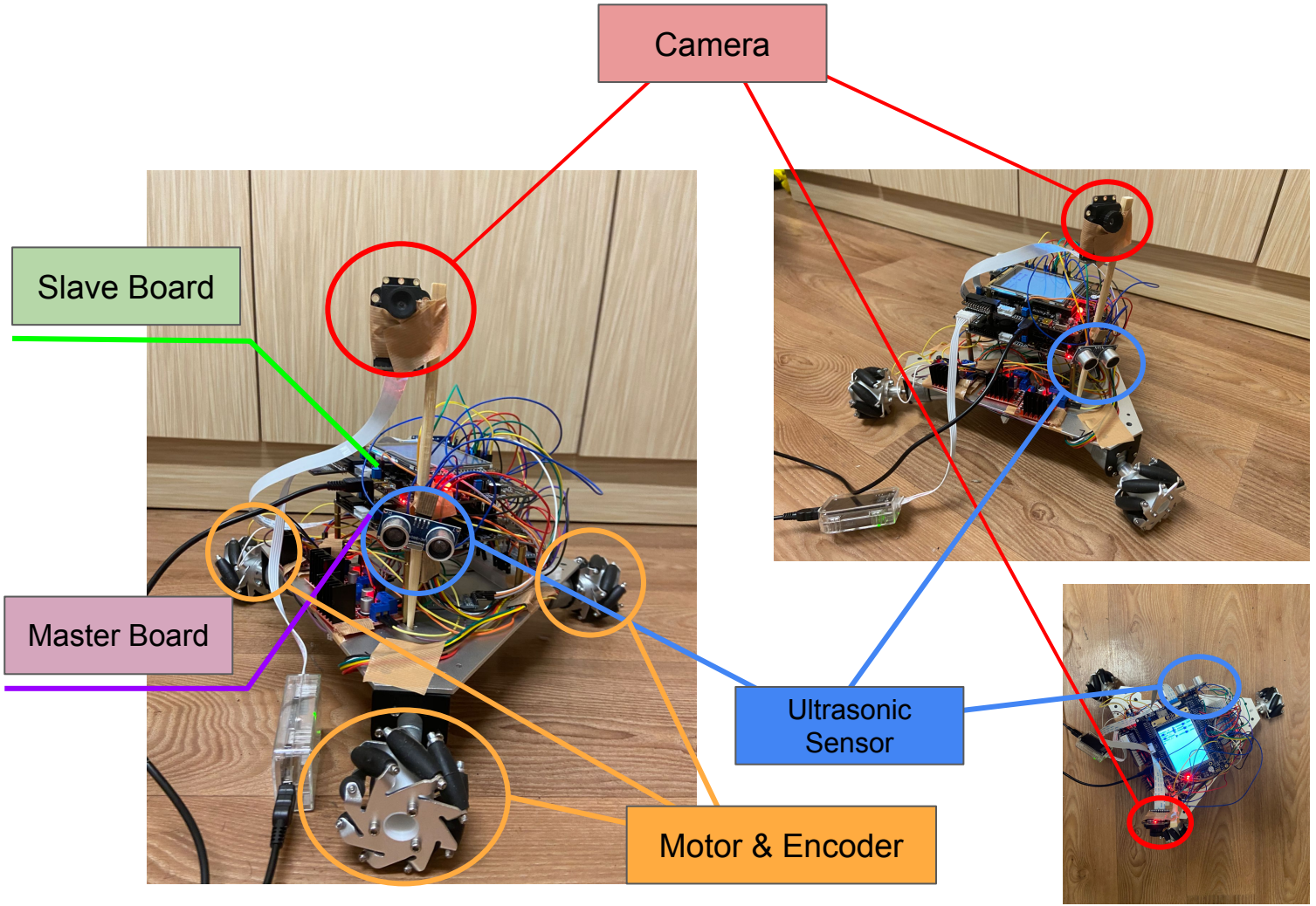
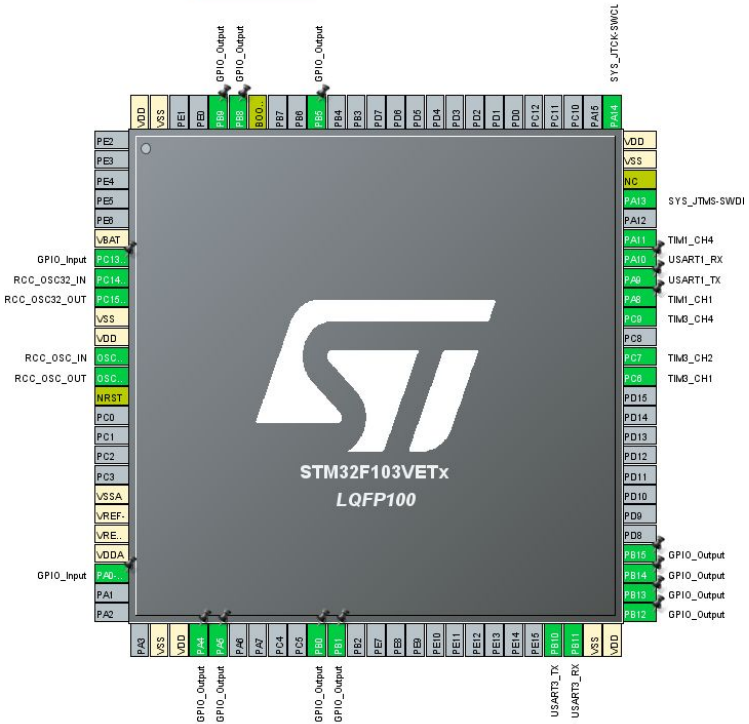
Group 18

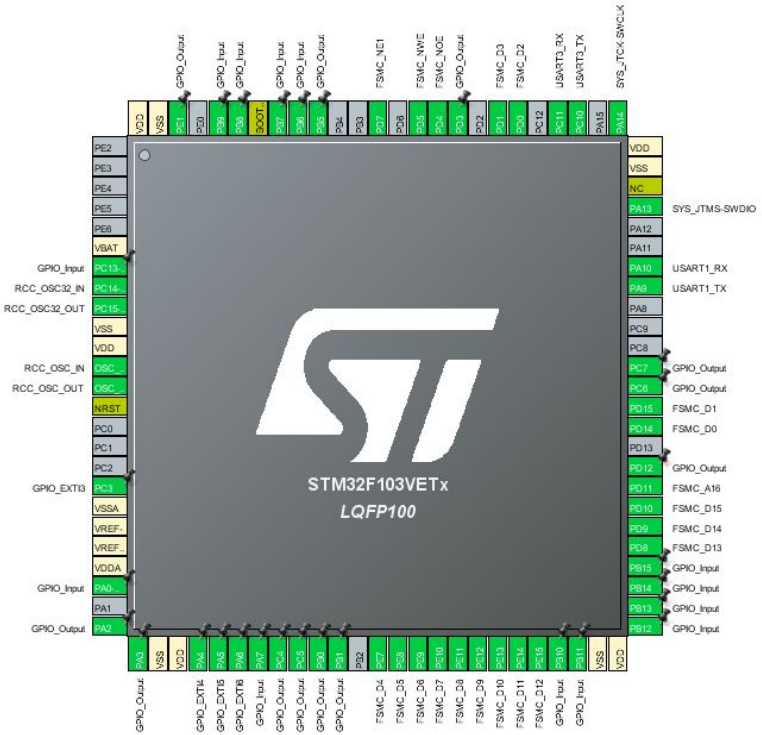
Yih CHENG
Mu-Ruei TSENG


Camera
Slave Board
Master Board
Ultrasonic Sensor
Motor & Encoder

Pin Layout for the two boards



Master Board



Slave Board

Master Board

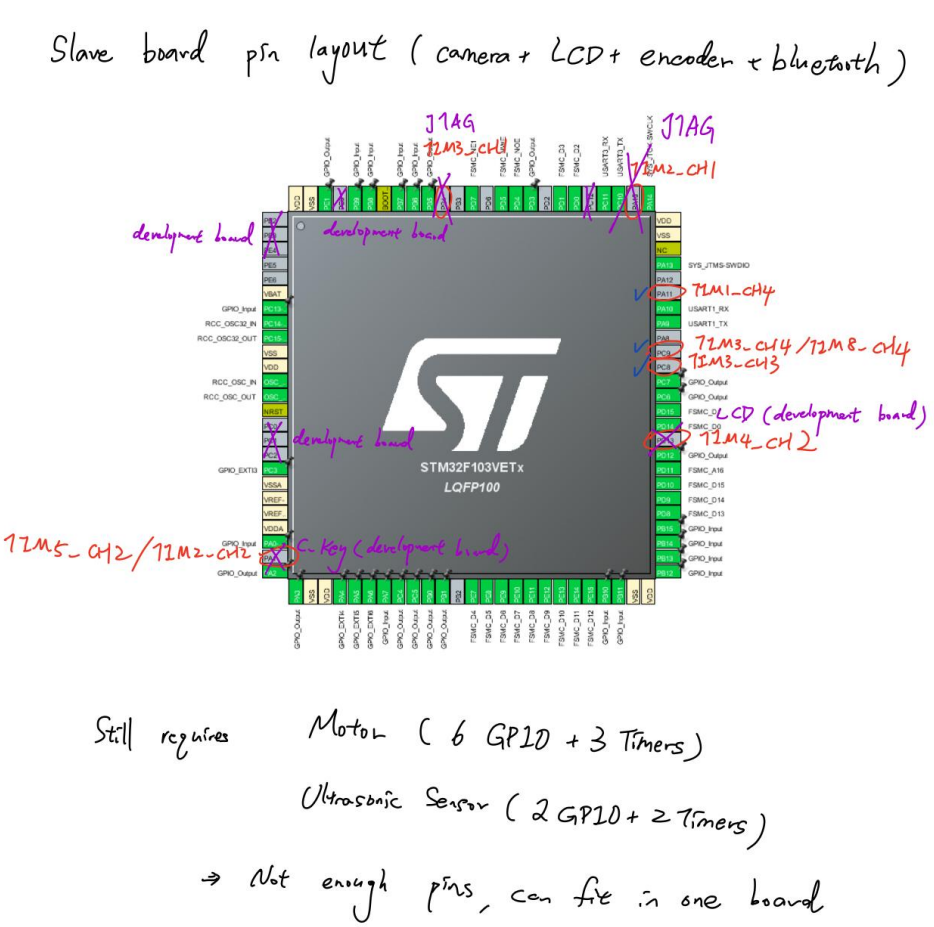
- Set motors' pwm and control the motors based on values calculated from the PC (GUI).
- Obtain distance from the ultrasonic sensor and stop the car when the car is reaching the wall.
- Send motor speed information to the slave board to display on the LCD

Slave Board

- Get camera image and encoder values
- Display necessary information on the LCD

Reasons for not using just one board

1. Not enough available pins
2. Doesn't have enough transmitting power when sending both the real-time image and also control data
3. The workload for sending encoder values, image pixels, and receiving PWM signals all in real time is too large for only one board.



Communication

- We use bluetooth and USART for communication
  - Bluetooth:
    - PC -> Master Board (PWM)
    - Slave Board -> PC (Camera image, encoder values)
  - USART:
    - Communication between master and slave board
- Bluetooth Packaging
  - Insert header to determine package type
  - Apply size check to prevent package loss

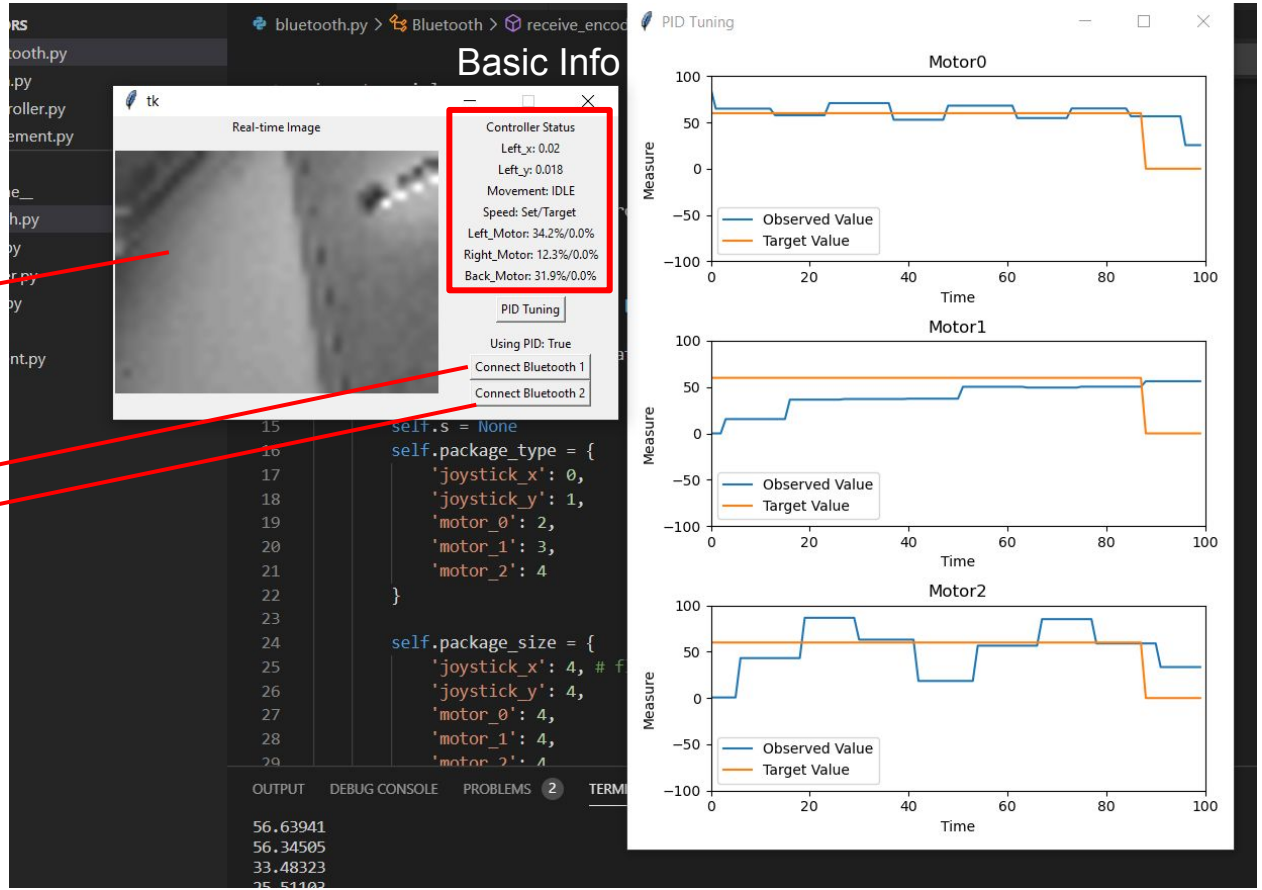
User Interface (GUI)

- Get the signal from Xbox controller, use it to set the movement of the car.
- Pop-up window showing the enocder value of the three motors.
- Calculate PID based on the encoder values to maintain the motor speed at a desire speed.
- Display images captured by the camera
- Buttons to connect/reconnect bluetooth

User Interface (GUI)

Real-time Image

Button to connect the bluetooths



PID Control

Reason: Adjust the motor pwm signal to reach so that the motor can operate in the desire rpm we set.

We use encoders' values to compare with the target value we set to determine the error that can be used as the input for the controller to adjust the output speed.

PID control:
P: Proportional (Kp)
Output is proportion to the error multiply a constant Kp.
I: Intergral (Ki)
Integral path will continue sum up the errors as the time move on and multiply with a constant Ki. It can be used to remove constant error since no matter how small the constant error is, the summation of the error can be significant enough to adjust the controller output.
D: Derivative (Kd)
Detect the rate of changes in the error. The faster the error changes, the larger the D value will output.

Camera

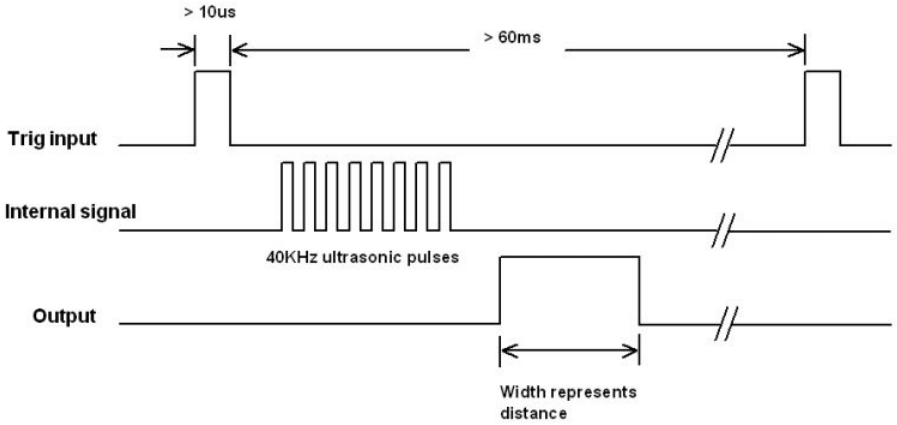
- Location: Slave Board
- Image size: 60x80 (height x width)
- Frame rate: 1~2 frames per sec
- Grayscale
- Function: Observe the environment
- The captured image is sent from from the slave board to the GUI on PC

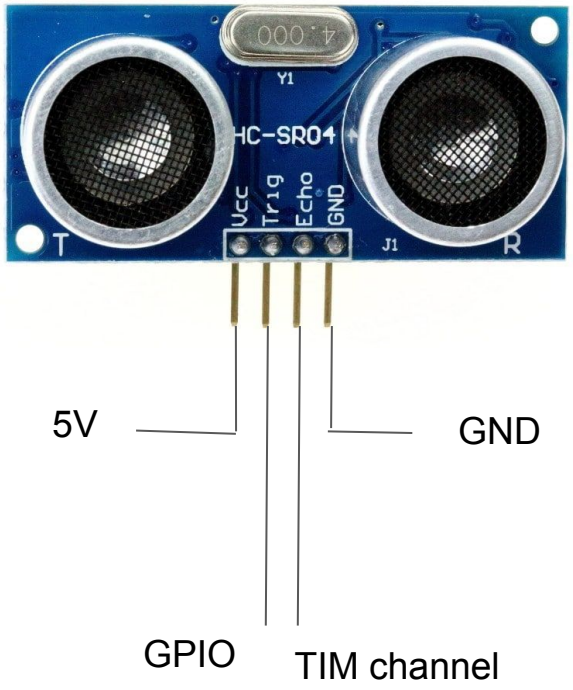
Camera Code Implementation

1. Camera Sensors capture image Pixels.
2. Camera then writes the pixels into FIFO buffer
3. Different Clock signals of OV7725 in order to control writing buffer and reading buffer time.
4. STM32 reads the FIFO for camera data.
5. STM32 sends these camera data to computer through Bluetooth (HC-05).

Ultrasonic Sensor

- Location: Master Board
- Detect distance between the car and and walls/obstacles
- Function: Stop the car when the distance is within 15 cm





1. TRIG pin High for 10us.
2. HC-SR04 send out 8 cycles of 40kHz ultrasound
3. Signal returns, and will output High pulse.

Distance = Output High level time * speed (340m/s) / 2

Problems

1. Camera color error
2. The receive/transmit time for Bluetooth is slow, causing a large delay in the system.
3. We use two boards because we don’t have enough pins available for interrupt (encoder).

Demo link

https://www.youtube.com/watch?v=Du-H7-a1xOw&ab_channel=YihCHENG