Technical Update Details

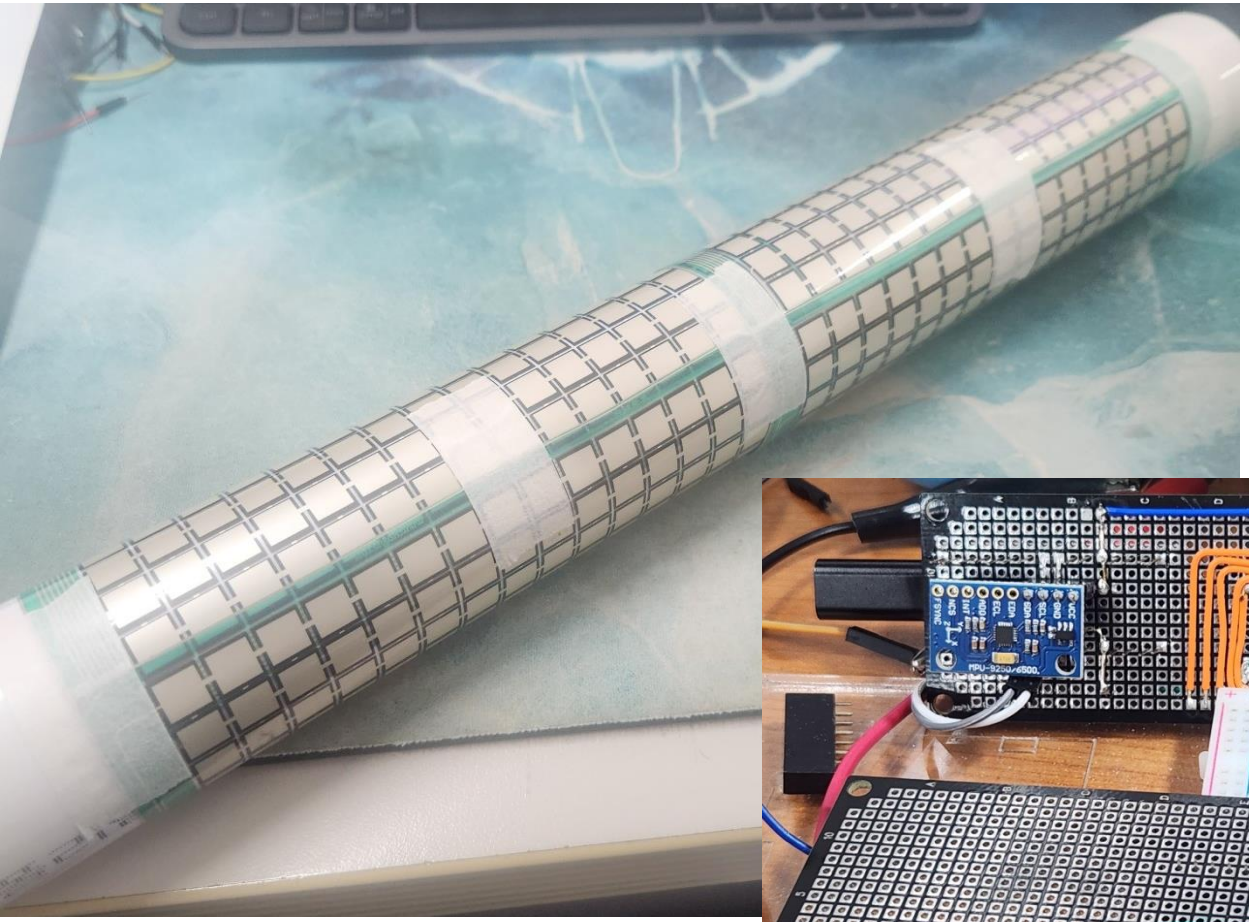
- About Sensor

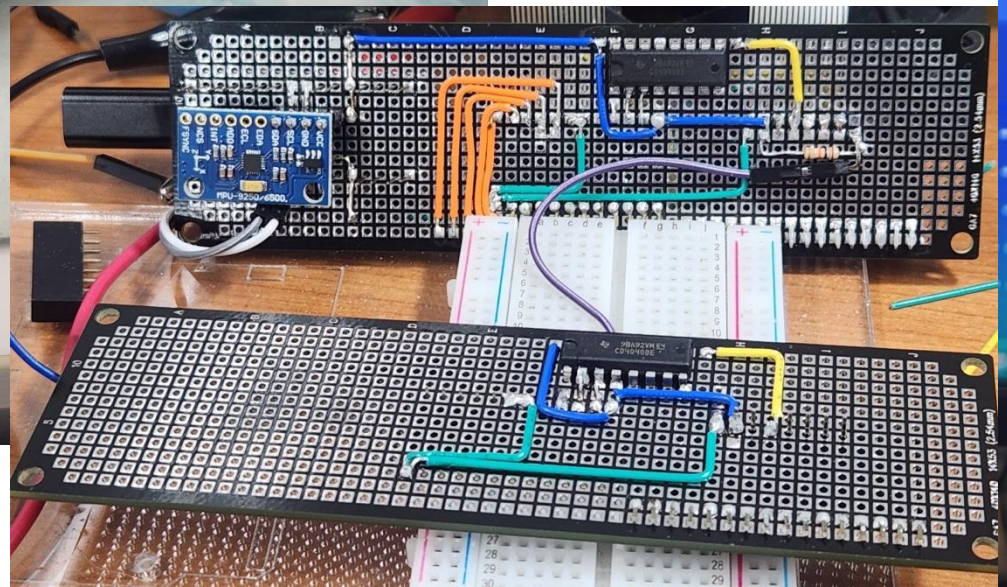
Capstone Design 2025-1, Team A#

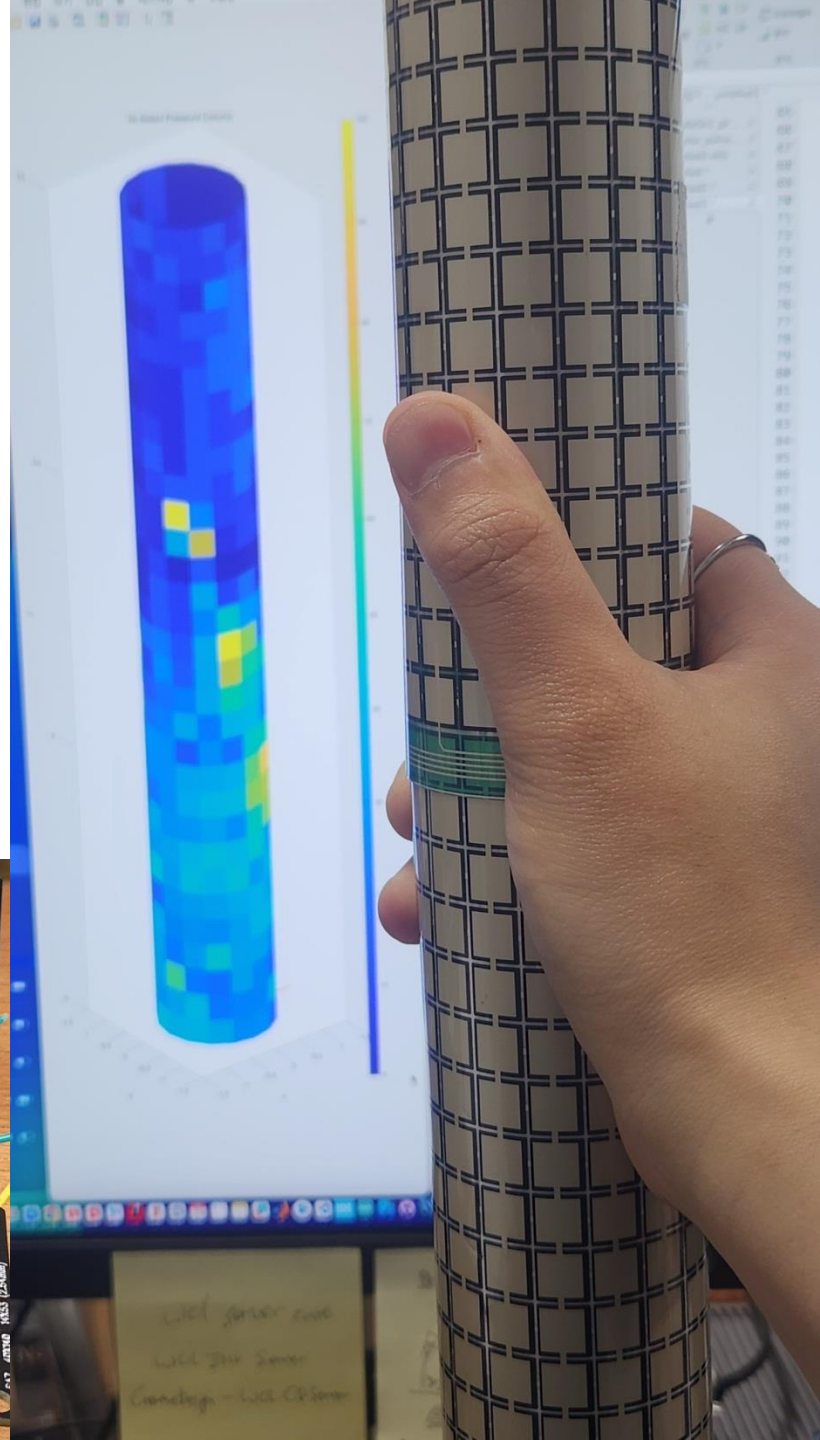
2025.05.01 Week 09

Presenter : 이진성

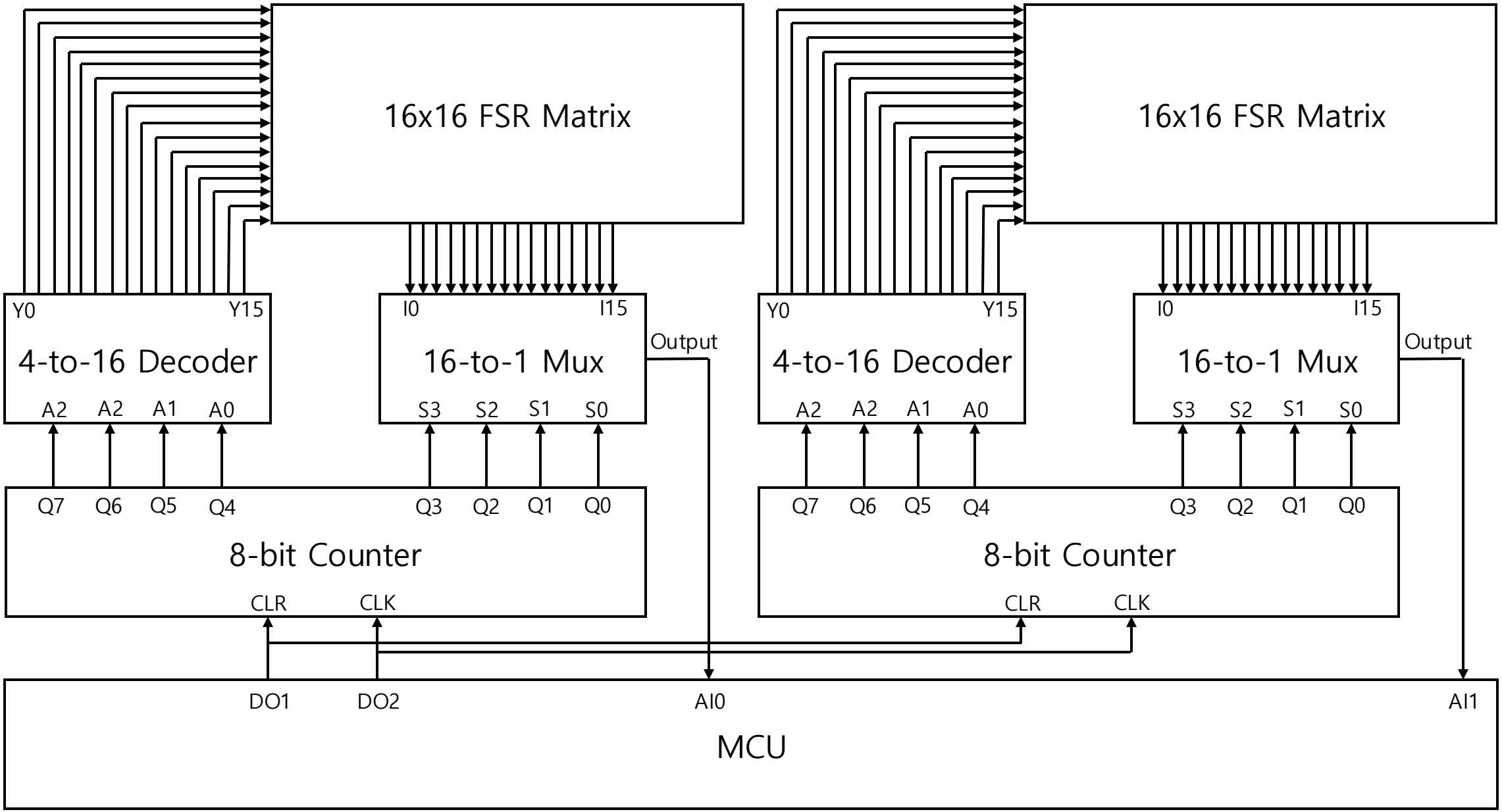
1. FSR Bar



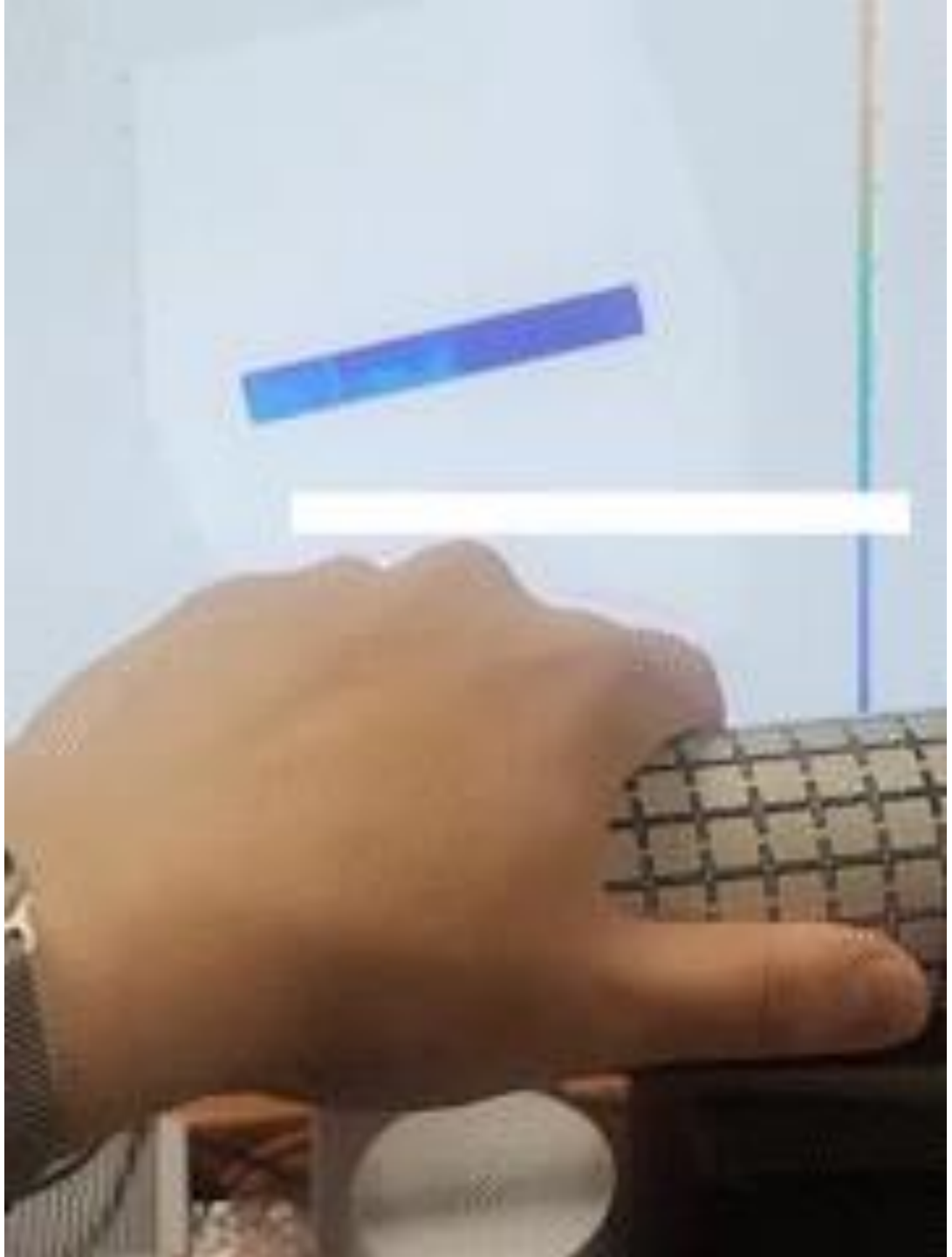




Dual FSR Matrix sensor

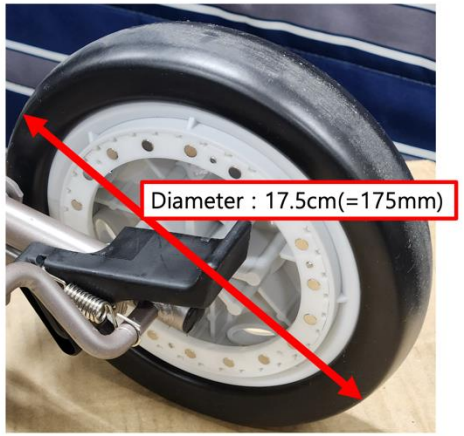


Gyro Sensor



Wheel Speed

Wheel Rotation and Displacement



$$Diameter = 17.5cm = 175mm$$

$$Circumference = Diameter \times \pi\ mm \approx 549.78mm$$

$$Two\ Magnet\ Distance = \frac{Circumference}{18(Magnetic)} \approx 30.5mm$$

Speed Calculation Using Timer1

- System Clock – 16MHz(16,000,000Hz)
- Timer1 is 16bit Counter – Maximum is 65,535(=2^{16}-1)

→Prescaler is 63(=64-1)

1. **CNT값의 증가량 확인:**
   - 16MHz 클럭에서 prescaler가 63이므로, 타이머의 클럭은 다음과 같이 계산됩니다:

$$= \frac{16,000,000}{63+1} = 250,000\ \text{Hz}$$

즉, 타이머는 250kHz로 카운트를 증가하게 됩니다.

2. **타이머 주기 계산:**
   - 타이머 클럭이 250kHz이므로 타이머의 주기는 다음과 같습니다:

$$= \frac{1}{250,000} = 4\ \mu s$$

3. **CNT값을 통한 시간 계산:**
   - 두 번의 interrupt 사이에서 CNT 값의 증가량을 ΔCNT라고 할 때, 이 증가량에 따른 경과 시간은 다음과 같습니다:

$$= \Delta CNT \times 4\ \mu s$$

4. **이동 속도 계산:**
   - 장치가 이 경과 시간 동안 3.05cm를 이동했다고 주어졌으므로, 이동 속도는 다음과 같이 계산할 수 있습니다:

$$(\text{cm/s}) = \frac{(\text{cm})}{(\text{s})} = \frac{3.05}{\Delta CNT \times 4 \times 10^{-6}}$$

   - 이를 단순화하면:

$$(\text{cm/s}) = \frac{3.05 \times 10}{4 \times \Delta CNT} = \frac{762,500}{\Delta CNT}$$

EXTI8 ISR and Calculate Speed

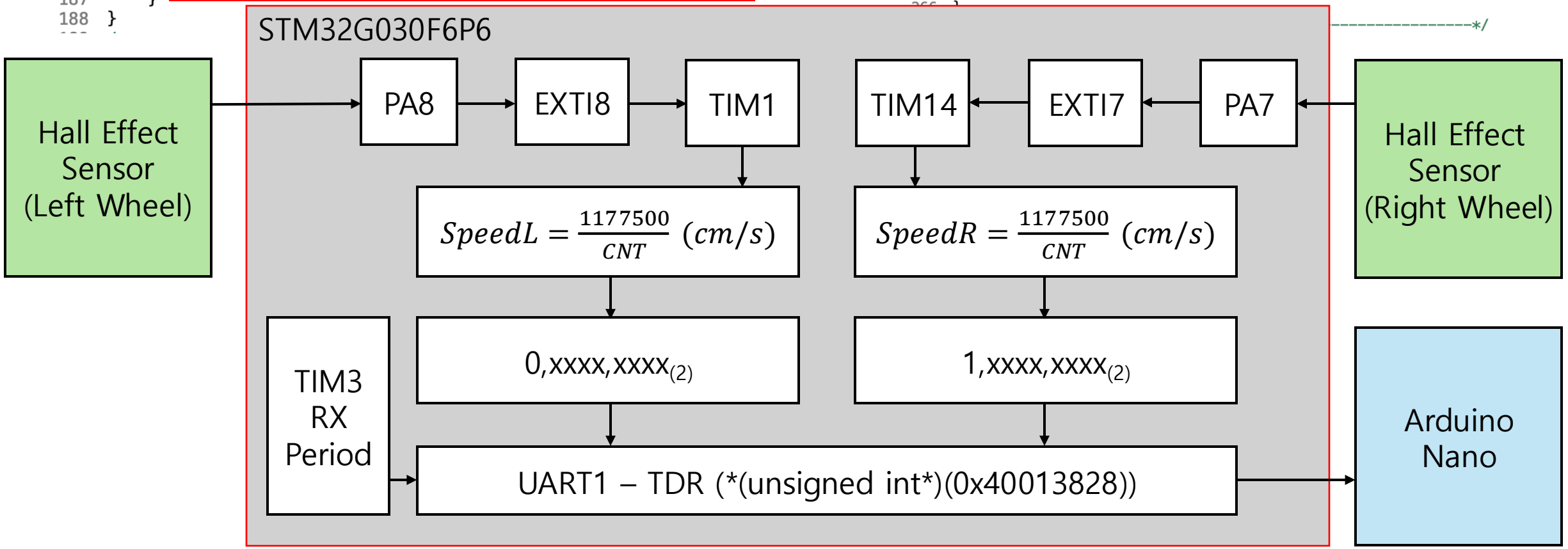
```
174 void EXTI4_15_IRQHandler(){
175     if(PCVUIP(0x4002180C) & (0x01<<8)){
176
177         PCVUIP(0x4002180C) |= 0x01<<8;
178
179         unsigned int cnt = PCVUIP(0x40012C24) & 0xFFFF;
180         PCVUIP(0x40012C24) = 0x00;  //copy CNT value and reset
181         speed = (unsigned char)(762500/cnt);
182 //      PCVUIP(0x40013828) = speed;
183
184         ledDebug();
185     }
186 }
```

```
if(PCVUIP(0x4002180C) & (0x01<<8)){

    PCVUIP(0x4002180C) |= 0x01<<8;

    unsigned int cnt = PCVUIP(0x40012C24) & 0xFFFF;
    PCVUIP(0x40012C24) = 0x00;  //copy CNT value and reset
    speedL = (unsigned char)(1177500/cnt);
    PCVUIP(0x40013828) = speed;

    ledDebug();
}
```

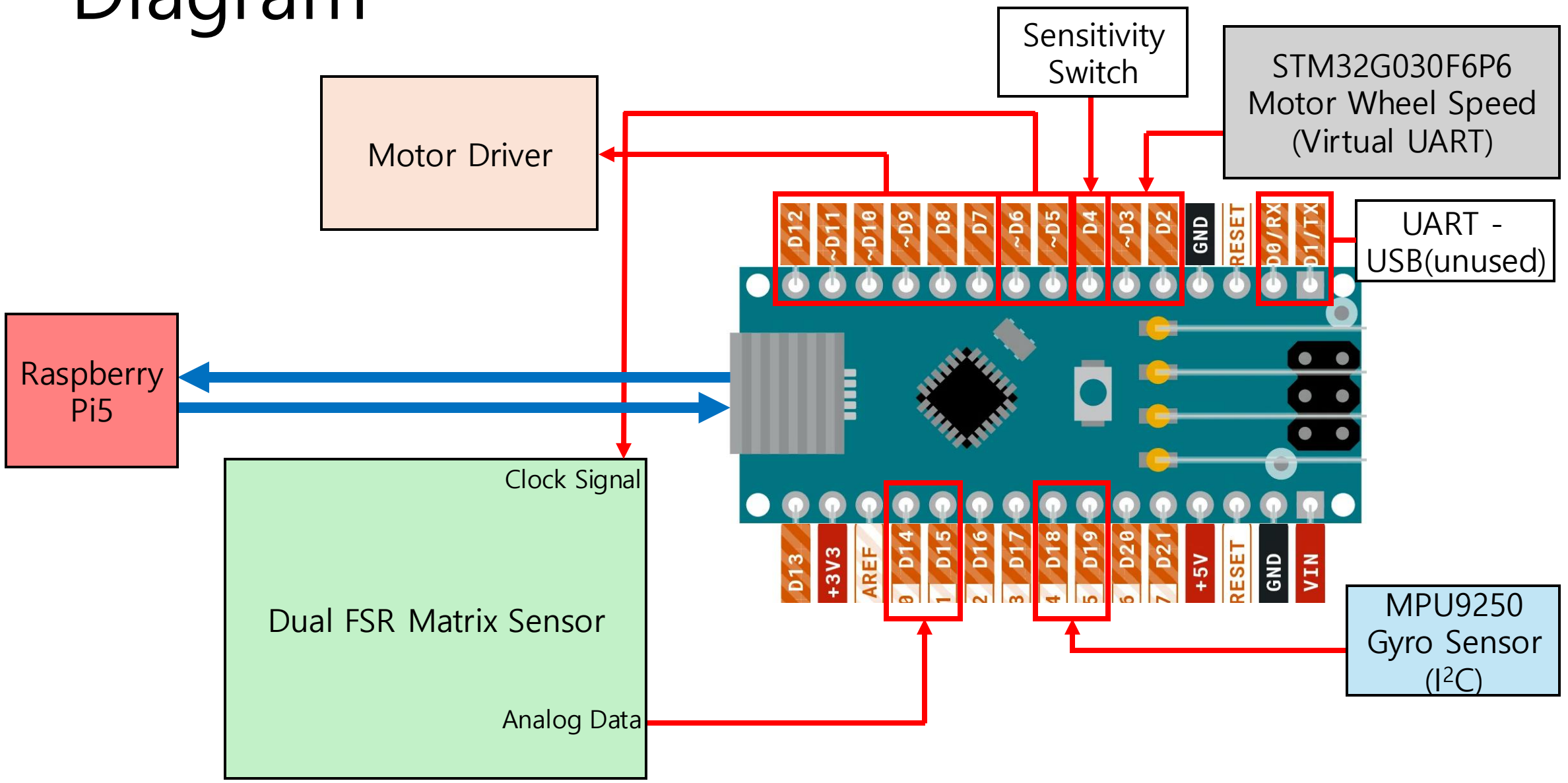
Dual Wheel Speed

```
179 void TIM3_IRQHandler(){
180     if(PCVUIP(0x40000410) & (0x01<<0)){
181         PCVUIP(0x40000410) &= ~(0x01<<0);
182
183         tx_dir ^= 1;
184         if(tx_dir) speed = speedR & 0xFF;
185         else       {speed = speedL & 0xFF;}
186         PCVUIP(0x40013828) = speed | (tx_dir<<8);    //UART TX
187     }
188 }
```

```
248 /*-------------------------------------------------------*/
249 void UART1Setting(){
250     PCVUIP(0x40021040) |=  0x01<<14;    //RCC uart1 enable
251     PCVUIP(0x40021034) |=  0x01<<1; //RCC GPIOB clock enable
252
253     PCVUIP(0x50000400) |=  (0x02<<(6*2));   //MODER
254     PCVUIP(0x50000400) &= ~(0x01<<(6*2));   //MODER
255     PCVUIP(0x50000400) |=  (0x02<<(7*2));   //MODER
256     PCVUIP(0x50000400) &= ~(0x01<<(7*2));   //MODER
257
258     PCVUIP(0x50000420) &= ~(0x0F<<(7*4));   //AFRL
259     PCVUIP(0x50000420) &= ~(0x0F<<(6*4));   //AFRL
260
261     PCVUIP(0x40013800) |=  (0x01<<12);      // 1 baud - 9 bit
262
263     PCVUIP(0x4001380C)  =  1667;    //baud rate : 16000000/9600
264     PCVUIP(0x40013800) |=  0x01<<0; //UART Enable
265     PCVUIP(0x40013800) |=  0x01<<3; //UART Transmitter Enable
266 }
                                        ----------------*/
```



Diagram


Sensitivity Switch
STM32G030F6P6 Motor Wheel Speed (Virtual UART)
Motor Driver
UART - USB(unused)
Raspberry Pi5
Clock Signal
Dual FSR Matrix Sensor
Analog Data
MPU9250 Gyro Sensor (I²C)
D12
~D11
~D10
~D9
D8
D7
~D6
~D5
D4
~D3
D2
GND
RESET
D0/RX
D1/TX
D13
+3V3
AREF
D14
D15
D16
D17
D18
D19
D20
D21
+5V
RESET
GND
VIN

Json Formating

```
292  void loop() {
293    sensingMPU9250();
294    readPressureData();
295    readWheelSpeed();
296
297    sendJsonStart();
298    sendMPU9250DataAsJson();
299    sendComma4Field2Field();
300    sendPressureDataAsJson();
301    sendComma4Field2Field();
302    sendCalibswitchJson();
303    sendComma4Field2Field();
304    sendWheelSpeedDataAsJson();
305    sendJsonEnd();
```

```
77   void sendMPU9250DataAsJson(){
78     // JSON 전송
79     Serial.print("\"MPU9250\":{");
80
81     Serial.print("\"accel\":{");
82     Serial.print("\"x\":"); Serial.print(ax, 3); Serial.print(",");
83     Serial.print("\"y\":"); Serial.print(ay, 3); Serial.print(",");
84     Serial.print("\"z\":"); Serial.print(az, 3); Serial.print("},");
```

```
145  void sendPressureDataAsJson() {
146    Serial.println("\"FSR\": {");
147
148    for (int row = 0; row < numRows; row++) {
149      Serial.print("\"row");
150      Serial.print(row);
151      Serial.print("\": [");
152
153      for (int col = 0; col < numCols * 2; col++) {
154        Serial.print(pressureData[row][col]);
155        if (col < numCols * 2 - 1) Serial.print(",");
```

```
250  void sendWheelSpeedDataAsJson(){
251    Serial.print("\"Wheel_Speed\": {\"L\":");
252    Serial.print(speedL);
253    Serial.print(", \"R\":");
254    Serial.print(speedR);
255    Serial.println("}");
```

Json Formating

```
51          % JSON 파싱
52          try
53              data = jsondecode(jsonStr);
54              if isfield(data, 'FSR')
55                  fsrData = data.FSR;
```


```
{
    "MPU9250": {
        "accel": {"x": -0.023, "y": 0.152, "z":  9.782},
        "gyro":  {"x": -0.032, "y": 0.041, "z": -0.003},
        "mag":   {"x":  12.45, "y": -7.32, "z":  42.13},
        "roll": 1.23, "pitch": -3.45, "yaw": 87.65
    },
    "FSR": {
        "row0": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12, 3
        "row1": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12, 3
        "row2": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12, 3
        "row3": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12, 3
        "row4": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12, 3
        "row5": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12, 3
        "row6": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12, 3
        "row7": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12, 3
        "row8": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12, 3
        "row9": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12, 3
        "row10": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12,
        "row11": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12,
        "row12": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12,
        "row13": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12,
        "row14": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12,
        "row15": [12, 34, 56, 78, 90, 12, 34, 56, 78, 90, 12,
    },
    "calib_switch": 0,
    "Wheel_Speed": { "L": 78, "R": 65 }
}
```


```python
def parse_serial_line(line, context):
    try:
        data = json.loads(line)
    except json.JSONDecodeError:
        print("[Error] JSON 디코딩 실패:", line)
        return None

    # 자이로 센서 기울기값
    mpu_data = data.get("MPU9250", {})
    context["pitch"] = mpu_data.get("pitch", 0.0)

    # FSR 데이터 파싱
    fsr_data = data.get("FSR", {})
    matrix = []
    for row_idx in range(16):
        row_key = f"row{row_idx}"
        row_data = fsr_data.get(row_key, [0]*32)
        matrix.append(row_data)

    context["fsr_matrix"] = matrix

    # calib_switch 데이터 파싱
    context["calib_switch"] = data.get("calib_switch", 0)

    # 🆕 Wheel_Speed 데이터 파싱
    wheel_data = data.get("Wheel_Speed", {})
    context["speedL"] = wheel_data.get("L", 0)
    context["speedR"] = wheel_data.get("R", 0)
    print(f"[수신] Wheel_Speed - L: {context['speedL']}, R: {context['speedR']}")
```

Thank you