Smart Software Project

Lab: Week 12

Gyro, Accelerometer & Compass Sensors

Prof. HyungJune Lee

hyungjune.lee@ewha.ac.kr

이화여자대학교
EWHA WOMANS UNIVERSITY

Today

- Lab announcement
- SmartCAR Gyro + Accelerometer sensors
- Lab assignment #9
- Course announcement

Class Schedule

Week	Lecture Contents	Lab Contents
Week 1	Course introduction	Arduino introduction: platform & programming environment
Week 2	Embedded system overview & source management in collaborative repository (using GitHub)	Lab 1: Arduino Mega 2560 board & SmartCAR platform
Week 3	ATmega2560 Micro-controller (MCU): architecture & I/O ports, Analog vs. Digital, Pulse Width Modulation	Lab 2: SmartCAR LED control
Week 4	Analog vs. Digital & Pulse Width Modulation	Lab 3: SmartCAR motor control (Due: HW on creating project repository using GitHub)
Week 5	ATmega2560 MCU: memory, I/O ports, UART	Lab 4: SmartCAR control via Android Bluetooth
Week 6	ATmega2560 UART control & Bluetooth communication between Arduino platform and Android device	Lab 5: SmartCAR control through your own customized Android app (Due: Project proposal)
Week 7	Midterm exam	
Week 8	ATmega2560 Timer, Interrupts & Ultrasonic sensors	Lab 6: SmartCAR ultrasonic sensing
Week 9	Infrared sensors & Buzzer	Lab 7: SmartCAR infrared sensing
Week 10	Acquiring location information from Android device & line tracing	Lab 8: Implementation of line tracer
Week 11	Gyroscope, accelerometer, and compass sensors	Lab 9: Using gyroscope, accelerometer, and compass sensors
Week 12	Project	Team meeting (for progress check)
Week 13	Project	Team meeting (for progress check)
Week 14	Course wrap-up & next steps	
Week 15	Project presentation & demo I (Due: source code, presentation slides, & poster slide)	Project presentation & demo II
Week 16	Final week (no final exam)	

Lab Session

- Practice in-lab programming exercises based on the lecture materials

- Upload source codes for lab assignments in Ewha Cyber Campus after the lab session
  - Due: 11:59pm on the lab day

- Once you are done, you can leave the session after checking with me or TA

- Or, continue to work on programming for other homework assignments

Lab Policy

- 1) Please check out your SmartCAR (& Nexus 7 tablet) as soon as you arrive at the classroom
- 2) Please complete lab assignments
- 3) Upload required files to Ewha Cyber Campus
- 4) Check with me or TA
- **5) Please upload a null firmware to SmartCAR before you return it!!!**
  - **This will be a part of your lab score**
- 6) Please remove files that you created or downloaded in your computer after you are done
  - Remove your project completely
- 7) Please shut down your computer before you leave
- 8) Return the checked-out SmartCAR (& Nexus 7 tablet) to TA

NOTE: How to run SmartCAR in Lab

- Power OFF
  - Compile your code
  - Lift up your SmartCAR with your hand
  - Upload your code
- Disconnect the USB cable
- Go to find a spacious area
- Put it down there
- Power ON
- It will run your firmware
- After test, turn power OFF

Lab Announcement

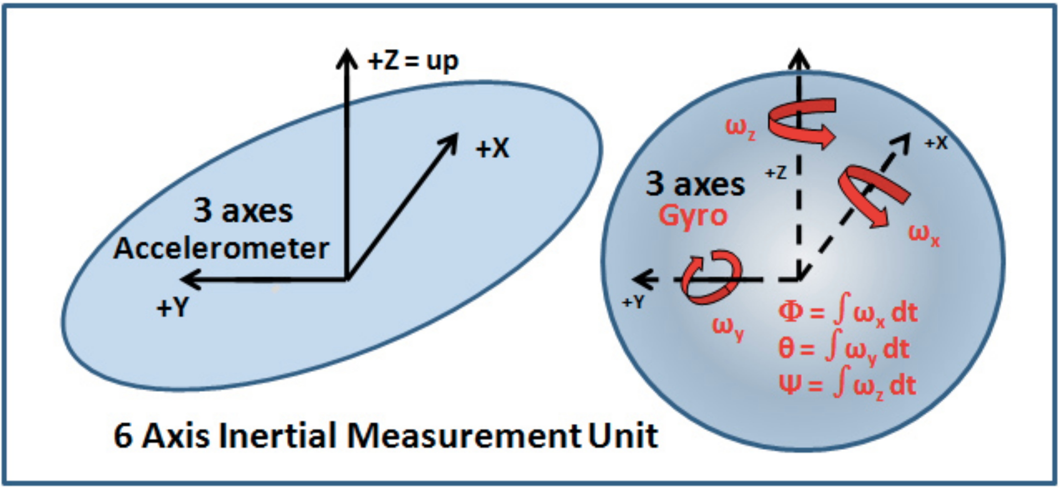
- Bluetooth pairing “headache”
  - Because there are so many Bluetooth devices in the classroom with the same name
    - “**155v2.1.7_hb**” <- SmartCAR
  - So please go outside with your SmartCAR and your Android device, and then pair them
  - Please do not pair with other students’ devices

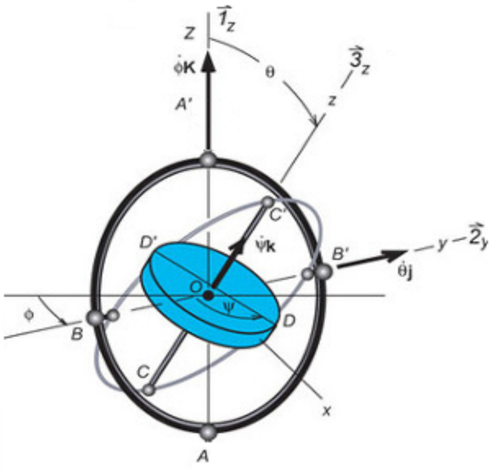
Today

- Lab announcement
- **SmartCAR Gyro + Accelerometer**
- Lab assignment #11
- Course announcement

3 axes Accelerometer + 3 axes Gyro

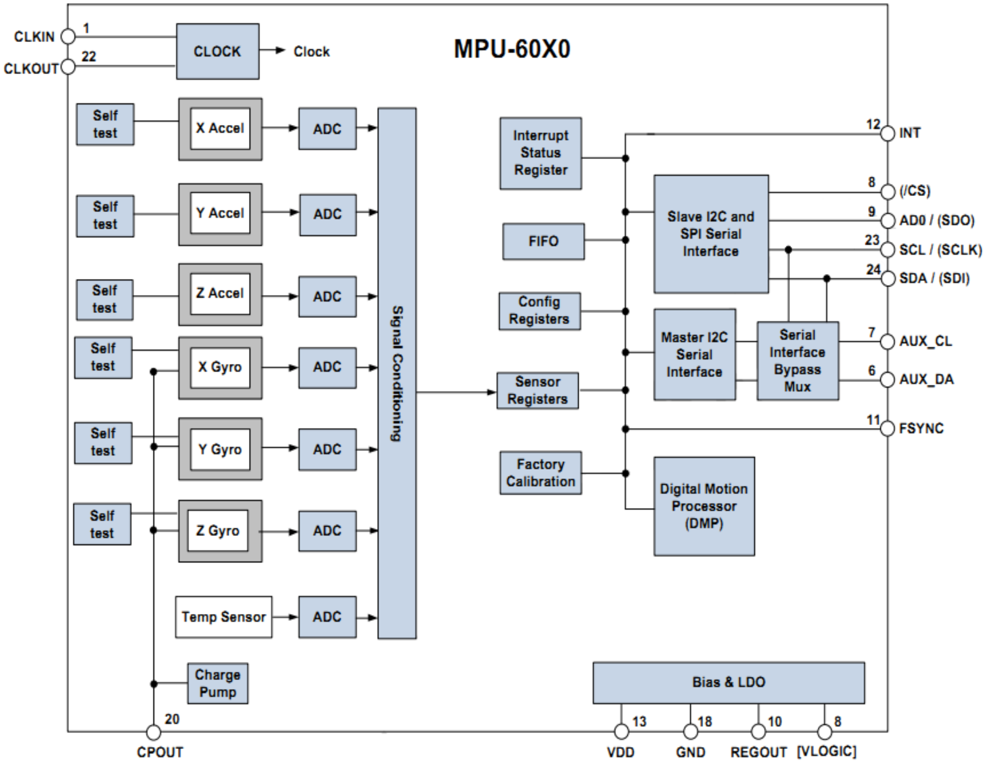
- 6 axes
  - 1) from 3 axes accelerometer
    - Measure proper acceleration in three axes
  - 2) from 3 axes gyroscope
    - Measure orientation, based on the principles of angular momentum

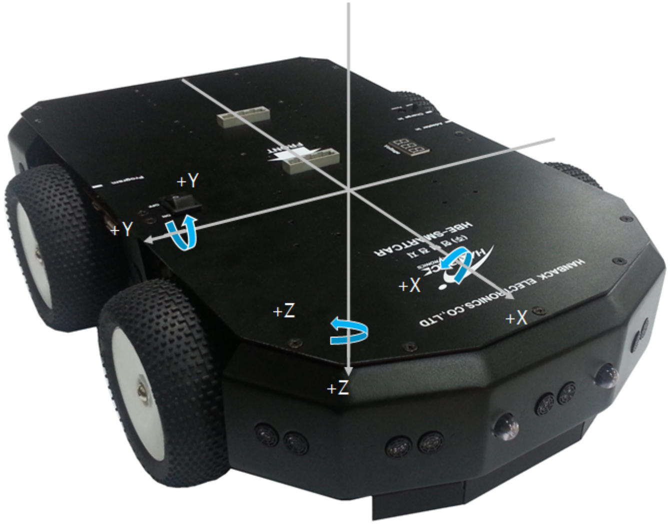

+Z = up
+X
3 axes
Accelerometer
+Y
3 axes
Gyro
+Z
+X
ω_z
ω_x
ω_y
+Y
$\Phi = \int \omega_x \, dt$
$\theta = \int \omega_y \, dt$
$\Psi = \int \omega_z \, dt$
6 Axis Inertial Measurement Unit



6 Axes Sensors in SmartCAR

- InvenSense MPU-6050: Gyro 3 axis + Accelerometer 3 axis sensors
  - Data transmission type to Atmega MCU
    - I^{2}C(Inter-Integrated Circuit) interface

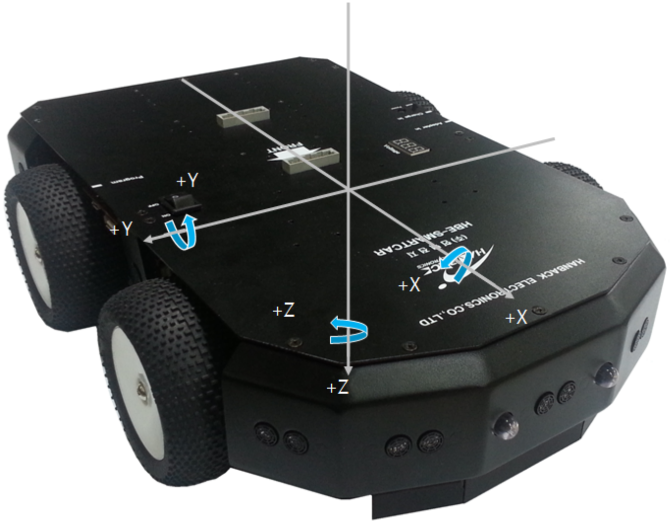




Yaw, pitch, and roll calculation

- Let’s have the followings from 6 axes measurements

  - Yaw: rotation in z-axis
  - Pitch: rotation in y-axis
  - Roll: rotation in x-axis



Part I: SmartCAR Firmware

- Lab11.h

```
#ifndef lab11_H_
#define lab11_H_
#include "Arduino.h"

#include "Wire.h"
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"

#ifdef __cplusplus
extern "C" {
#endif
void loop();
void setup();
#ifdef __cplusplus
}
#endif

#endif
```

Part I: SmartCAR Firmware

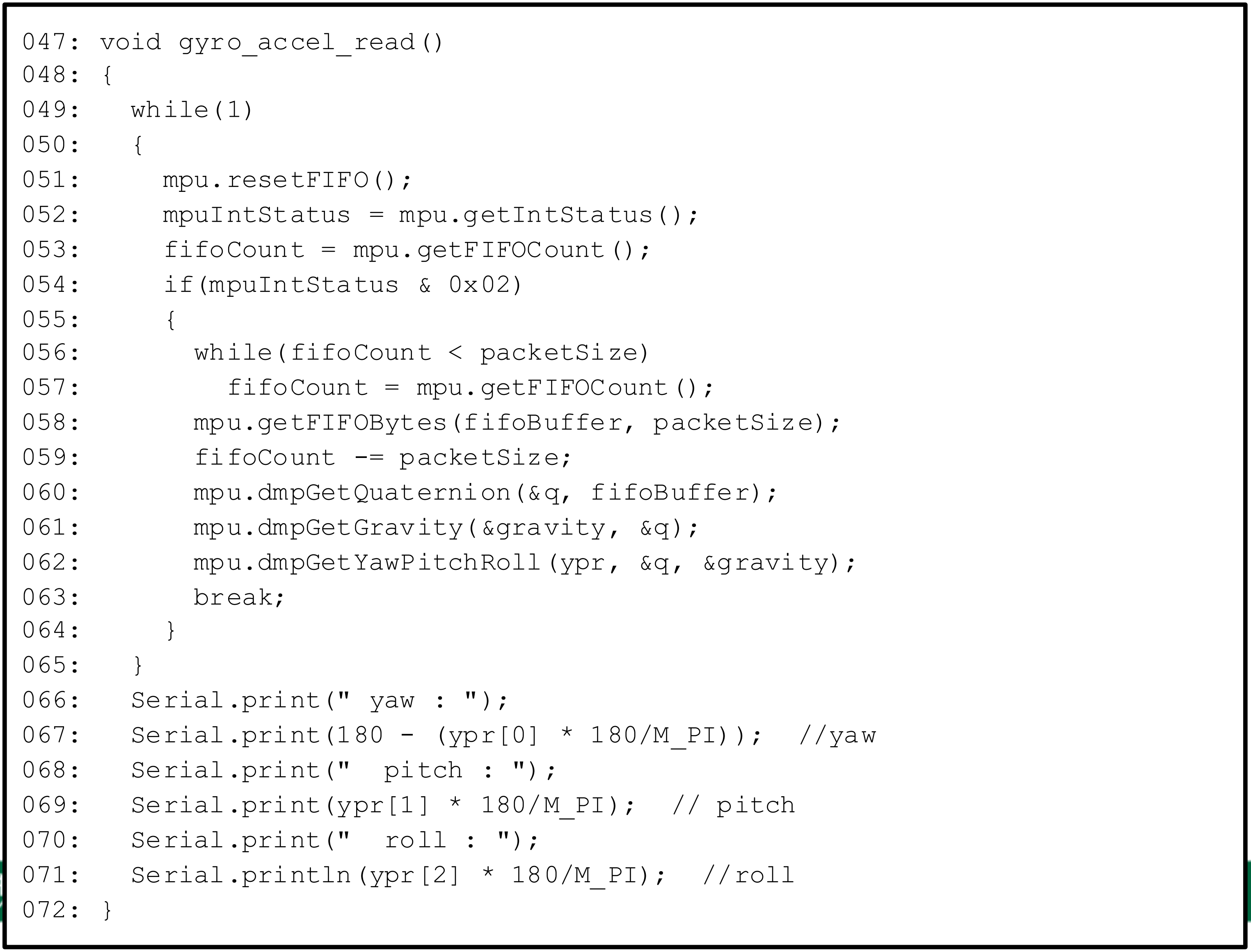
- lab11.cpp

```
#include "lab11.h"

MPU6050 mpu;

uint8_t mpuIntStatus;   // holds actual interrupt status byte from MPU
uint8_t devStatus;      // return status after each device operation (0 = success, !0 = error)
uint16_t packetSize;    // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount;     // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

Quaternion q;           // [w, x, y, z]         quaternion container
VectorFloat gravity;    // [x, y, z]            gravity vector
float ypr[3];           // [yaw, pitch, roll]   yaw/pitch/roll container and gravity vector

void setup()
{
  Wire.begin();

  Serial.begin(115200);

  Serial.println("Initializing I2C devices…");
  mpu.initialize();
```

Part I: SmartCAR Firmware

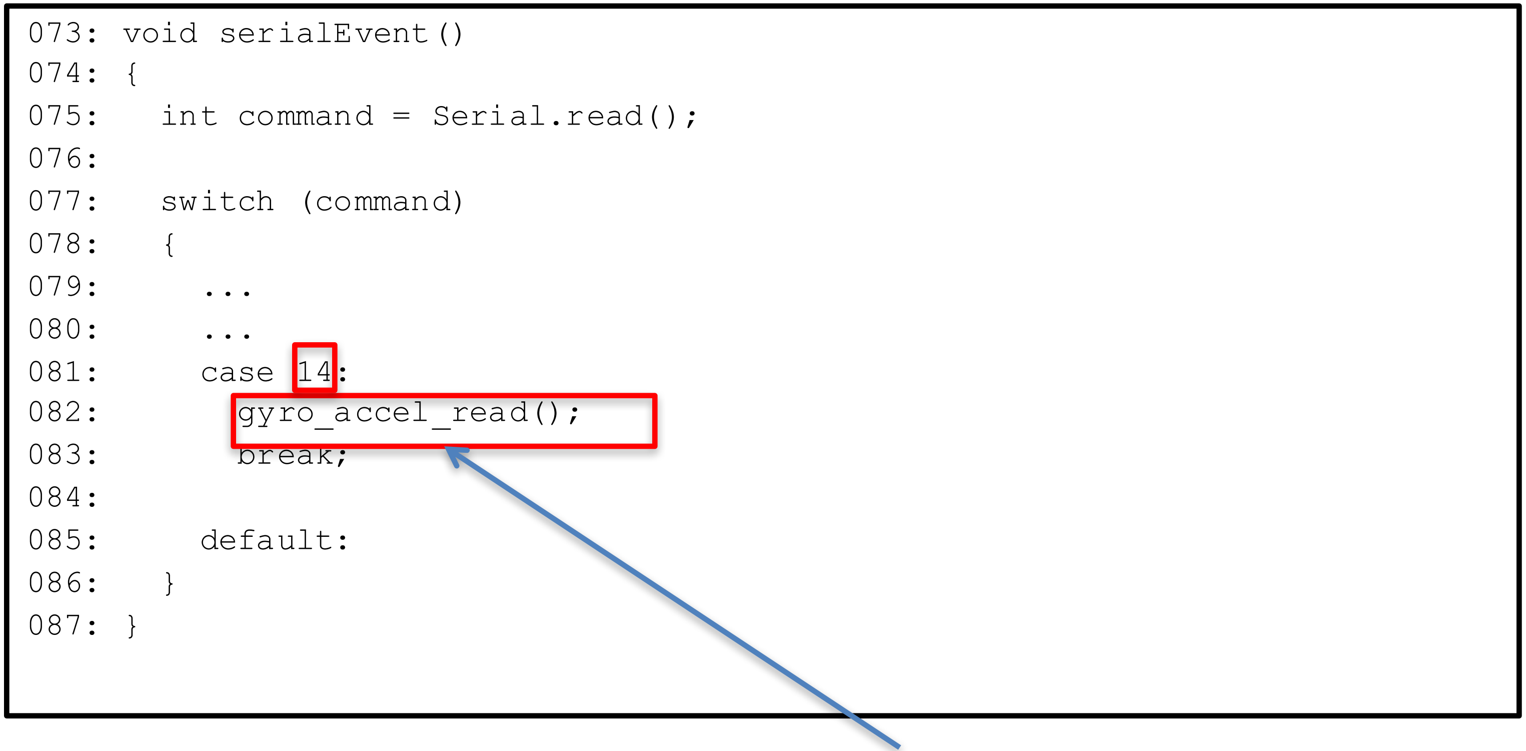
```

  Serial.println("Testing device connections…");
  Serial.println(mpu.testConnection() ? "MPU6050 connection
successful"          :  "MPU6050 connection failed");

  Serial.println("Initializing DMP…");
  devStatus = mpu.dmpInitialize();

  if (devStatus == 0) {

    Serial.println("Enabling DMP…");
    mpu.setDMPEnabled(true);

    packetSize = mpu.dmpGetFIFOPacketSize();
  }
  else
  {
    Serial.print("DMP Initialization failed (code ");
    Serial.print(devStatus);
    Serial.println(")");
  }
}
void loop()
{
}
```

Part I: SmartCAR Firmware



```
void gyro_accel_read()
{
  while(1)
  {
    mpu.resetFIFO();
    mpuIntStatus = mpu.getIntStatus();
    fifoCount = mpu.getFIFOCount();
    if(mpuIntStatus & 0x02)
    {
      while(fifoCount < packetSize)
        fifoCount = mpu.getFIFOCount();
      mpu.getFIFOBytes(fifoBuffer, packetSize);
      fifoCount -= packetSize;
      mpu.dmpGetQuaternion(&q, fifoBuffer);
      mpu.dmpGetGravity(&gravity, &q);
      mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
      break;
    }
  }
  Serial.print(" yaw : ");
  Serial.print(180 - (ypr[0] * 180/M_PI));  //yaw
  Serial.print("  pitch : ");
  Serial.print(ypr[1] * 180/M_PI);  // pitch
  Serial.print("  roll : ");
  Serial.println(ypr[2] * 180/M_PI);  //roll
}
```

Part I: SmartCAR Firmware



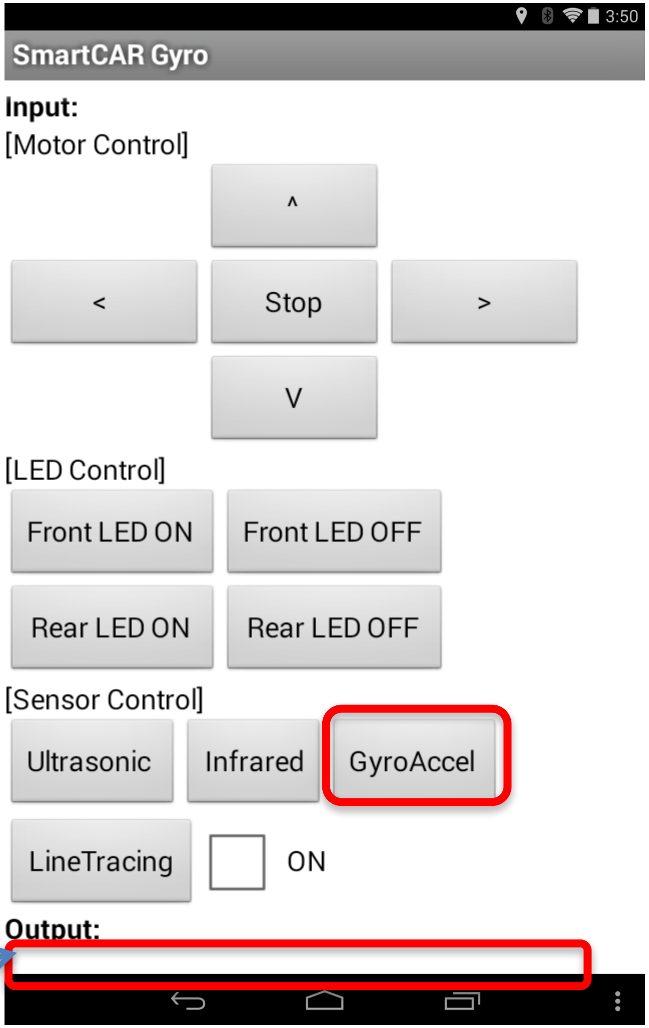
```
void serialEvent()
{
  int command = Serial.read();

  switch (command)
  {
    ...
    ...
    case 14:
      gyro_accel_read();
      break;

    default:
  }
}
```

- If the SmartCAR receives a byte of 14, read gyroscope + accelerometer sensors

Part II: SmartCAR Gyro App

- http://ai2.appinventor.mit.edu
- Click on “New Project”
- Enter “SmartCAR_Gyro” in Project Name
(One word, no space)

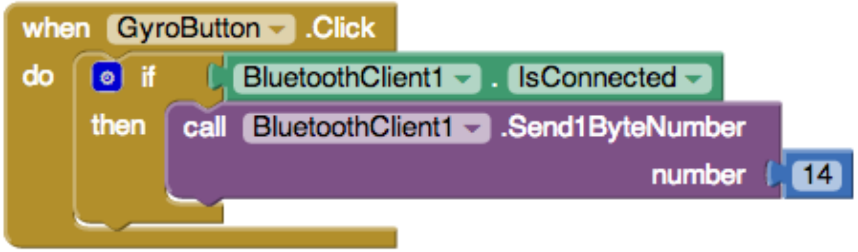
- Under “User Interface”
  - Drag-and-drop “Button” component
    - To send command
  - Drag-and-drop “CheckBox” component
    - Enable the line tracing, or
    - Disable the line tracing



BTOutput (Label)

SmartCAR Rotation Check

- Read measurements from Gyro and Accelerometer
  and print out in a label
  (command byte: 14)
  - Send "14" in number using "BluetoothClient.Send1ByteNumber"



```
when GyroButton .Click
do  if   BluetoothClient1 . IsConnected
    then call BluetoothClient1 .Send1ByteNumber
                                number 14
```

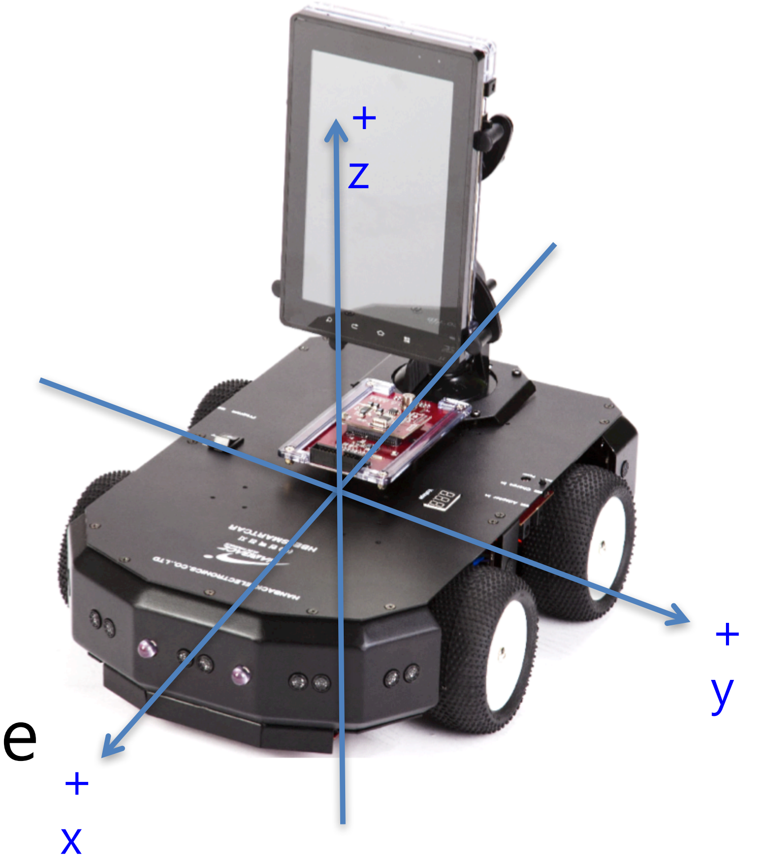
Lab Assignment #9

- Submit three following files to Cyber Campus
  - 1) lab9.cpp (Arduino firmware code)
  - 2) lab9.h (Arduino firmware code)
  - 3) SmartCAR_Gyro.apk (Android app package)
    - You should set the app icon image to “SmartCAR.png”
    - In App Inventor,
      “Build” ➔ “App (save .apk to my computer)”
  - 4) lab9_comment.txt
    - Explain what is going on these measurements while rotating

- Show your result to TA or instructor

Lab Assignment #9: lab9_comment.txt

- Please answer why the measurement is changing in that way... found out any rule?
  - When you rotate in x-axis
    - Clockwise
    - Counter clockwise
  - When you rotate in y-axis
    - Clockwise
    - Counter clockwise
  - When you rotate in z-axis
    - Clockwise
    - Counter clockwise



- Create lab9_comment.txt file to explain this

Course Announcement

- Next Week
  - Project discussion meetings: 10 – 15 minutes per team
  - Team 1 ~ 5