Smart Software Project

Lab: Week 11

Line Tracing with Infrared Sensors

Prof. HyungJune Lee
hyungjune.lee@ewha.ac.kr

이화여자대학교
EWHA WOMANS UNIVERSITY

Today

- Lab announcement
- Line tracing
- Lab assignment #8
- 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

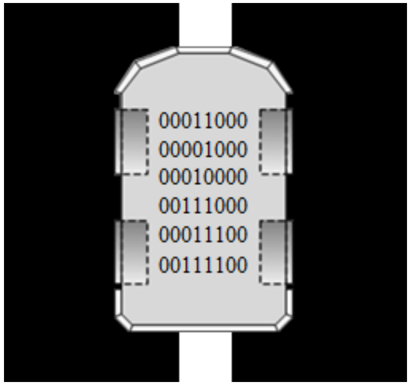
- Bluetooth pairing “headache”
  - Because there are so many Bluetooth devices in the classroom with the same name
    - “**155v2.1.7_hb**” <- SmartCAR
  - So please go outside with your SmartCAR and your Android device, and then pair them
  - Please do not pair with other students’ devices

Today

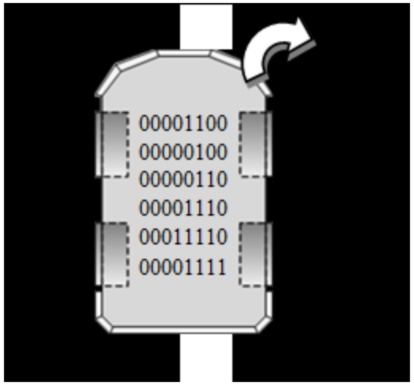
- Lab announcement
- **Line tracing**
- Lab assignment #8
- Course announcement

Line Tracer

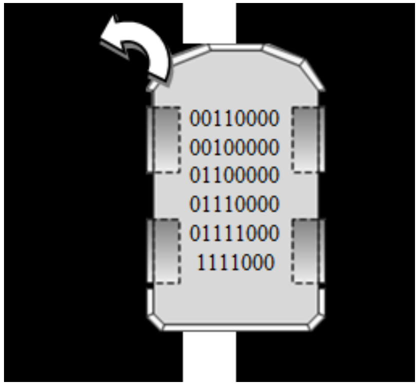
- Line tracing in SmartCAR
  - Infrared sensor data depending on SmartCAR's position



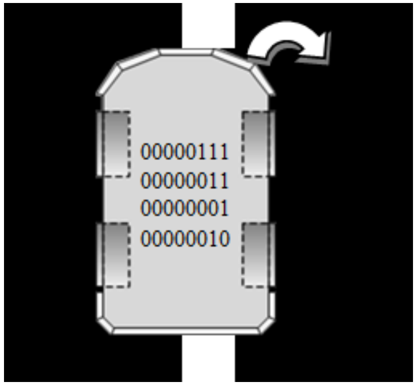
(a) Forward



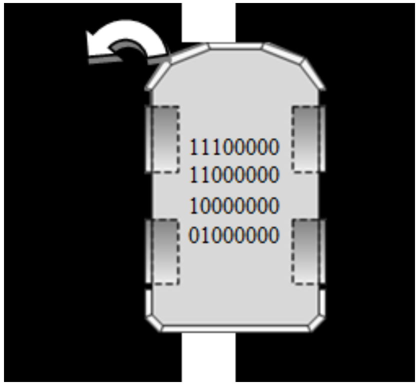
(b) Smooth Right-turn



(c) Smooth Left-turn

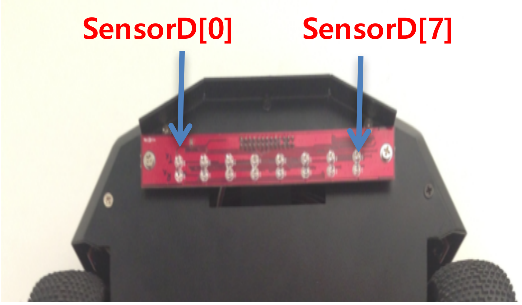


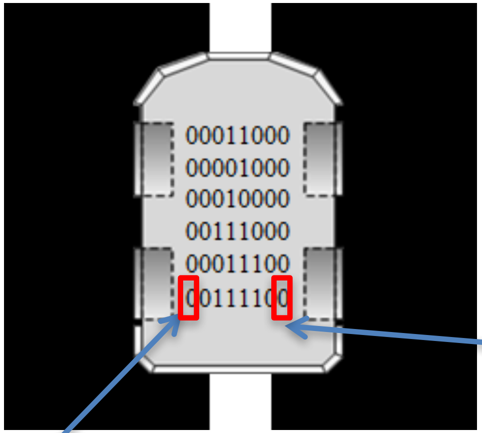
(d) Pivot Right-turn



(e) Pivot Left-turn

Sensor Data





```
…
…

unsigned char sensor_data = 0;
int z;

for(z=0;z<8;z++)
{
    unsigned int val = digitalRead(SensorD[z]);
    sensor_data |= (val << z);
}
```

SensorD[7]	SensorD[6]	SensorD[5]	SensorD[4]	SensorD[3]	SensorD[2]	SensorD[1]	SensorD[0]

To track black line in white background,

- we should complement the sensor_data ('1' to '0', '0' to '1')

```
sensor_data = ~sensor_data;
```

Control Motors w.r.t. Infrared Sensor

- How to control motors w.r.t. sensor_data

Sensor_data	Direction	Speed_data_L	Speed_data_R	Etc
0x18	FORWARD	140	140	Forward
0x10				
0x08				
0x38				
0x1C				
0x3C				
0x0C	RIGHT	200	0	Smooth Right Turn
0x04				
0x06				
0x0E				
0x1E				
0x0F				
0x30	LEFT	0	200	Smooth Left Turn
0x20				
0x60				
0x70				
0x78				
0xF0				
0x07	PIVOT_RIGHT	200	80	Pivot Right Turn
0x03				
0x02				
0x01				
0xC0	PIVOT_LEFT	80	200	Pivot Left Turn
0x40				
0x80				
0xE0				
0x00	STOP	0	0	Stop

Append your code to Lab 7

- When you receive a command byte to enable the line tracing mode,
  - `line_tracing = true;`
- When you receive a command byte to disable the line tracing mode,
  - `line_tracing = false;`
- In `loop()`, check the `line_tracing` flag
  - `if (line_tracing == true)`
    - Keep controlling the movement of SmartCAR

Part I: Start the line tracer

```
boolean line_tracing = false;

...

void line_tracing_enable()
{
    line_tracing = true;
    Serial.write("Line tracing is enabled..");
}

void line_tracing_disable()
{
    line_tracing = false;
    move_stop();
    Serial.write("Line tracing is disabled..");
}
```

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

    switch (command)
    {
        ...
        case 12:
            line_tracing_enable();
            break;

        case 13:
            line_tracing_disable();
            break;
        default:
    }
}
```

- If the SmartCAR receives a byte of 12, enable line tracer
- If the SmartCAR receives a byte of 13, disable line tracer

Part I: Start the line tracer

```
void loop()
{
    if (line_tracing == true)
    {
        unsigned char sensor_data = 0;
        int z;

        for(z=0;z<8;z++)
        {
            unsigned int val = digitalRead(SensorD[z]);
            sensor_data |= (val << z);
        }
        sensor_data = ~sensor_data;

        Serial.print(sensor_data, HEX);
        Serial.write(" ");

        switch (sensor_data)
        {
            case 0x18:
            case 0x10:
            case 0x08:
            case 0x38:
            case 0x1c:
            case 0x3c:
                move_forward_speed(140, 140);
                break;
```

```
            case 0x0c:
            case 0x04:
            case 0x06:
            case 0x0e:
            case 0x1e:
            case 0x0f:
                turn_right_speed(200, 0);
                break;

            case 0x30:
            case 0x20:
            case 0x60:
            case 0x70:
            case 0x78:
            case 0xf0:
                turn_left_speed(0, 200);
                break;

            case 0x07:
            case 0x03:
            case 0x02:
            case 0x01:
                turn_pivot_right_speed(200, 80);
                break;
```

Part I: Start the line tracer

```
            case 0xc0:
            case 0x40:
            case 0x80:
            case 0xe0:
              turn_pivot_left_speed(80,  200);
              break;

            case 0x00:
            case 0xff:
              move_stop();
              break;

            default:
              move_stop();
              break;
         }
         delay(5);
      }
}
```

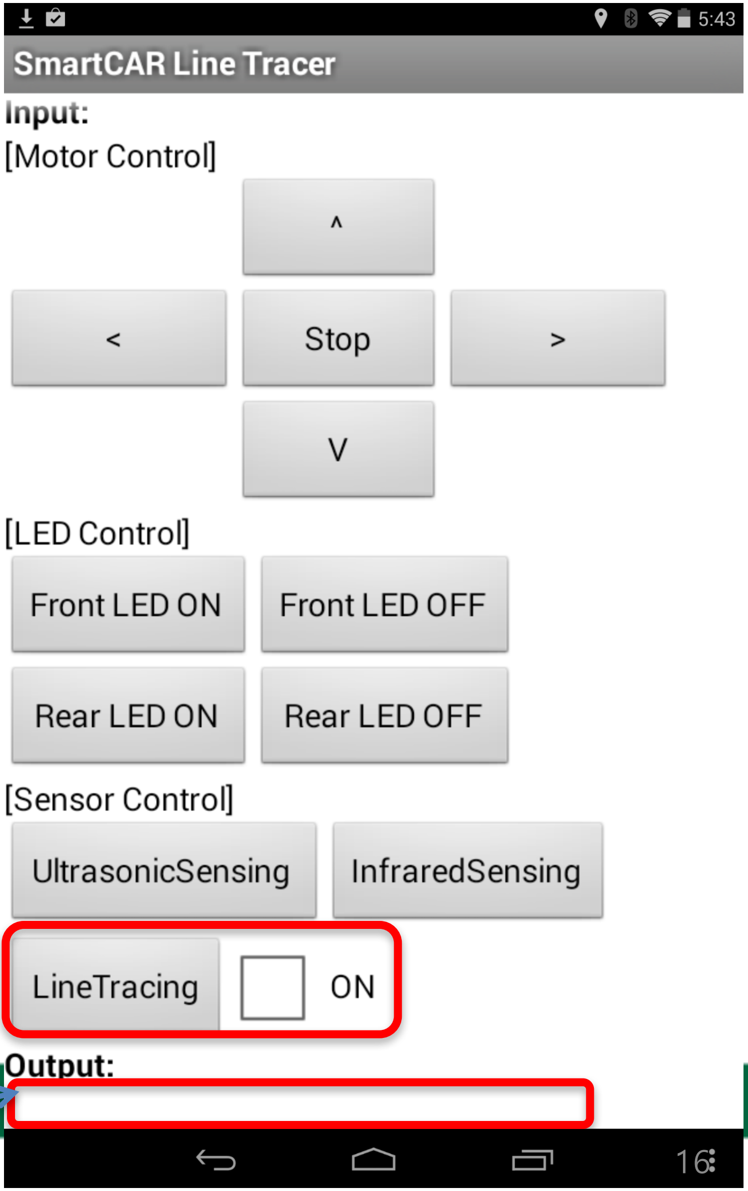
- Between each control,
  - Give a delay of 5ms
  - `delay(5);`
- Fill out the following functions

```
void move_forward_speed(int left, int right)
{
}
void turn_left_speed(int left, int right)
{
}
void turn_right_speed(int left, int right)
{
}
void turn_pivot_left_speed(int left, int right)
{
}
void turn_pivot_right_speed(int left, int right)
{
}
```

Part II: SmartCAR LT App

- http://ai2.appinventor.mit.edu
- Click on "New Project"
- Enter "SmartCAR_LT" 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

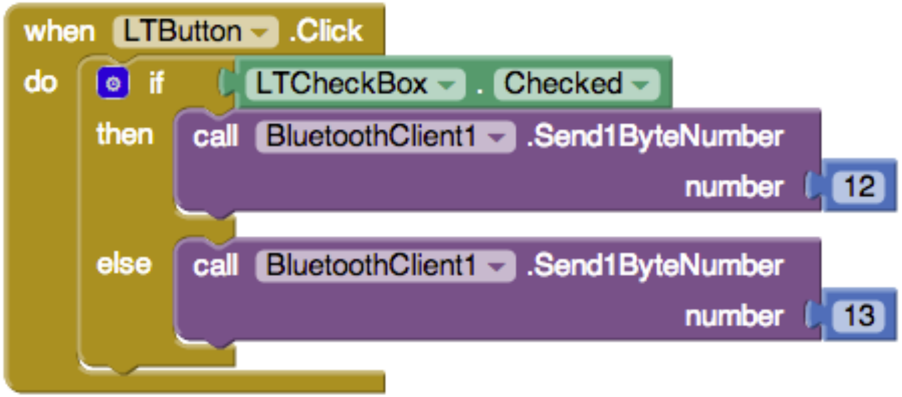


BTOutput (Label)

SmartCAR Line Tracing Control

- Enable the line tracer (command byte: 12)
  - Send “12” in number using “BluetoothClient.Send1ByteNumber”

- Disable the line tracer (command byte: 13)
  - Send “13” in number using “BluetoothClient.Send1ByteNumber”



```
when LTButton .Click
do  if    LTCheckBox . Checked
    then  call BluetoothClient1 .Send1ByteNumber
                                     number  12
    else  call BluetoothClient1 .Send1ByteNumber
                                     number  13
```

Lab Assignment #8

- Submit three following files to Cyber Campus
  - 1) lab8.cpp (Arduino firmware code)
  - 2) lab8.h (Arduino firmware code)
  - 3) SmartCAR_LT.apk (Android app package)
    - You should set the app icon image to “SmartCAR.png”
    - In App Inventor,
      “Build” ➔ “App (save .apk to my computer)”

- Show your result to TA or instructor

Course Announcement

- Next Week,
  - Using gyroscope sensor in SmartCAR
  - Using accelerometer sensor in SmartCAR
  - Using compass sensor in SmartCAR

- Next Lab session,
  - Applying gyroscope, accelerometer, and compass sensors