Smart Software Project

Lab: Week 4
SmartCAR
Motor Control

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

이화여자대학교
EWHA WOMANS UNIVERSITY

Today

- Lab announcement
- SmartCAR Motor Control
- Lab assignments #2-1, #2-2, #3
- 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

Lab Announcement

- Please review C programming language
- I posted a brief document that includes C essentials in Ewha Cyber Campus
- Without this fundamental, it is very difficult for you to learn new embedded programming skillsets

Lab Assignment #2-1

- Rear LED is initially ON (100%)
- After 2 seconds, rear LED's brightness is 75%
- After 2 seconds, rear LED's brightness is 50%
- After 2 seconds, rear LED's brightness is 25%
- After 2 seconds, rear LED is OFF

- Requirement) Use "digitalWrite()", not "analogWrite()"
- Hint) Use PWM
- 1) What happens when you set the cycle time to 100ms?
- 2) What happens as you decrease the cycle time to 50ms and 20ms?

- Upload your 1) lab2_1.h and 2) lab2_1.cpp file to Cyber Campus
- Show your result to TA or instructor

Lab Assignment #2-1

```
#define FRONT_LED_PIN          10
#define REAR_LED_PIN           9
#define DUTY_CYCLE             20    //20ms

int done = false;

void PWM_Write(int pin, int on_time_perc)
{
    int on_time = DUTY_CYCLE*on_time_perc/100.0;
    int off_time = DUTY_CYCLE - on_time;

    digitalWrite(pin, HIGH);
    delay(on_time);

    digitalWrite(pin, LOW);
    delay(off_time);

}

void myAnalogWrite(int pin, int percent, int time)
{
    int num_loops = time/DUTY_CYCLE;
    int i;
    for (i=0; i<num_loops; i++)
    {
        PWM_Write(pin, percent);
    }
}
```

```
//The setup function is called once at startup of
// the sketch
void setup()
{
    pinMode(REAR_LED_PIN, OUTPUT);
}


// The loop function is called in an endless loop
void loop()
{
    if (done == false) {
        myAnalogWrite(REAR_LED_PIN, 100, 2000);
        myAnalogWrite(REAR_LED_PIN, 75, 2000);
        myAnalogWrite(REAR_LED_PIN, 50, 2000);
        myAnalogWrite(REAR_LED_PIN, 25, 2000);
        done = true;
    }
    else
        digitalWrite(REAR_LED_PIN, LOW);
}
```

Lab Assignment #2-2

- Rear LED is initially ON
- Rear LED should be gradually darkening for 10 seconds and eventually OFF

- Requirement) Use "digitalWrite()", not "analogWrite()"
- Hint) Use PWM

- Upload your 1) lab2_2.h and 2) lab2_2.cpp file to Cyber Campus
- Show your result to TA or instructor

Lab Assignment #2-2

```
#define FRONT_LED_PIN          10
#define REAR_LED_PIN           9
#define DUTY_CYCLE             20    //20ms

int done = false;

void PWM_Write(int pin, int on_time_perc)
{
    int on_time = DUTY_CYCLE*on_time_perc/100.0;
    int off_time = DUTY_CYCLE - on_time;

    digitalWrite(pin, HIGH);
    delay(on_time);

    digitalWrite(pin, LOW);
    delay(off_time);

}

void myAnalogWrite(int pin, int percent, int time)
{
    int num_loops = time/DUTY_CYCLE;
    int i;
    for (i=0; i<num_loops; i++)
    {
        PWM_Write(pin, percent);
    }
}
```

```
//The setup function is called once at startup of the sketch
void setup()
{
    pinMode(REAR_LED_PIN, OUTPUT);
}

// The loop function is called in an endless loop
void loop()
{
    if (done == false)
    {
        int value = 100;
        int value_interval = 5;
        int total_duration = 10000;       //10,000ms

        int time_per_step = total_duration/(value/value_interval);

        while (value > 0)
        {
            myAnalogWrite(REAR_LED_PIN, value, time_per_step);
            value -= value_interval;
        }
        done = true;
    }
    else
        digitalWrite(REAR_LED_PIN, LOW);
}
```

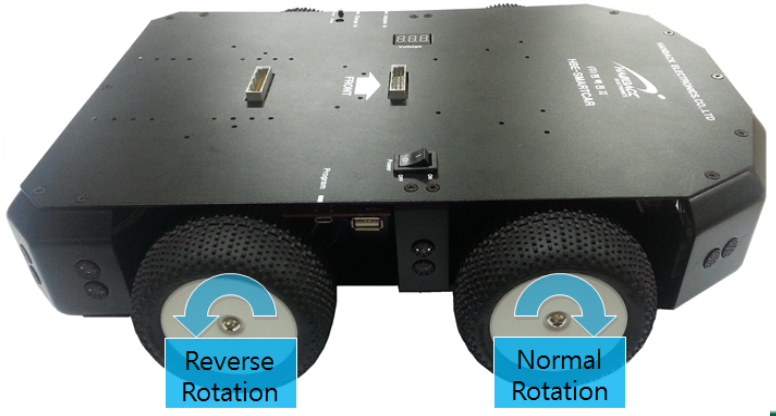
Today

- Lab announcement
- **SmartCAR Motor Control**
- Lab assignments #2-1, #2-2, #3
- Course announcement

SmartCAR Motor Control

- 4 motors
  - Two motors on the left operate together
  - Two motors on the right operate together
  - Controlled via PWM digital output from Arduino Mega 2560 board
  - 4 motions: 1) forward, 2) backward, 3) right, 4) left
  - Rear motors installed with encoders
    - You can check the number of rotations!

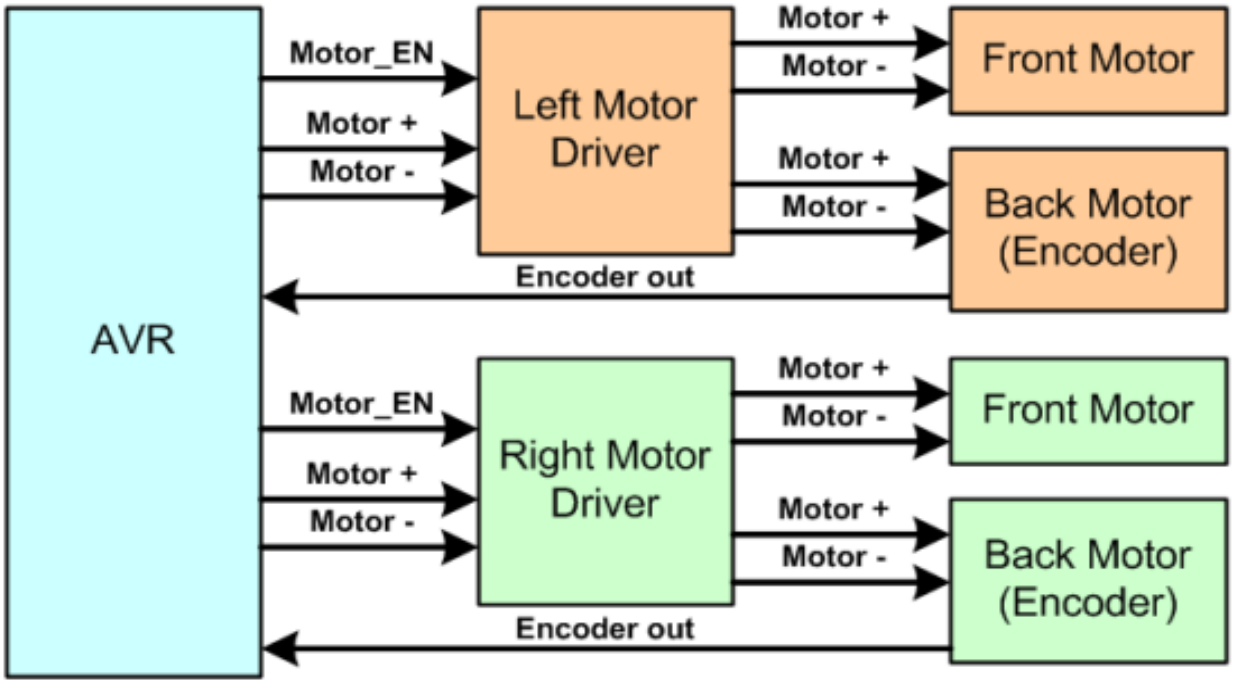
SmartCAR Motor Placement



SmartCAR Motor Rotation

DC Motor Operation

- Motor Driver: L298P chip
  - Motor Control Block diagram



SmartCAR Motor Port Table

Component	Name	Port / Number	Etc
MOTOR	LEFT_MD_A	PA0 / **22**	
	LEFT_MD_B	PA1 / **23**	
	RIGHT_MD_A	PA2 / **24**	
	RIGHT_MD_B	PA3 / **25**	
	L_MOTOR_EN	PC5 / **4**	
	R_MOTOR_EN	PE3 / **5**	
ENCODER	L_ENCOD_A	PE6 /	
	L_ENCOD_B	PK5 / A13	
	R_ENCOD_A	PE7 /	
	R_ENCOD_B	PK7 / A15	

SmartCAR Motor Port Table

Component	Name	Port / Number	Etc
MOTOR	LEFT_MD_A	PA0 / **22**	
	LEFT_MD_B	PA1 / **23**	
	RIGHT_MD_A	PA2 / **24**	
	RIGHT_MD_B	PA3 / **25**	
	L_MOTOR_EN	PC5 / **4**	
	R_MOTOR_EN	PE3 / **5**	
LED	FRONT_LED	PB4 / **10**	
	REAR_LED	PH6 / **9**	

SmartCAR Motor Control Value

- MOTOR_EN is connected to PWM output
  - MOTOR_EN: 0 (OFF) vs. 1(ON)
  - You can control motor speed by using analogWrite()

Control Signal			Operation	Etc
MD_A	MD_B	MOTOR_EN		
X	X	0	Stop	X -> 0 or 1
0	0	1	Stop	
0	1	1	Clockwise	
1	0	1	Counterclockwise	
1	1	1	Stop	

Port Control

- SmartCAR Motor Control
  - To decide CW or CCW, use MD_A and MD_B
    - "01" – rotating clockwise
    - "10" – rotating counterclockwise
  - To control rotation speed of motor, use MOTOR_EN
    - EN signal to '1': Motor is enabled
      - digitalWrite(pin, HIGH)
    - PWM output to EN signal for controlling speed
      - analogWrite(pin, value) where value is 0 to 255

Initialization in setup()

- Port 22, 23, 24, 25, 4, 5 should be initialized to
  - OUTPUT ports
  - LOW (OFF)

```
#define LEFT_MD_A       22
#define LEFT_MD_B       23
#define RIGHT_MD_A      24
#define RIGHT_MD_B      25

#define LEFT_MOTOR_EN 4
#define RIGHT_MOTOR_EN          5

//The setup function is called once at startup of the sketch
void setup()
{
    // Add your initialization code here
    pinMode(LEFT_MD_A, OUTPUT);
    pinMode(LEFT_MD_B, OUTPUT);
    pinMode(RIGHT_MD_A, OUTPUT);
    pinMode(RIGHT_MD_B, OUTPUT);
    pinMode(LEFT_MOTOR_EN, OUTPUT);
    pinMode(RIGHT_MOTOR_EN, OUTPUT);

    digitalWrite(LEFT_MD_A, LOW);
    digitalWrite(LEFT_MD_B, LOW);
    digitalWrite(RIGHT_MD_A, LOW);
    digitalWrite(RIGHT_MD_B, LOW);
    digitalWrite(LEFT_MOTOR_EN, LOW);
    digitalWrite(RIGHT_MOTOR_EN, LOW);
}
```

Moving forward

- To make SmartCAR move forward
- Left motors
  - Should be rotated
    - counterclockwise or clockwise?
    - Then LEFT_MD_A = ?, LEFT_MD_B = ?
- Right motors
  - Should be rotated
    - counterclockwise or clockwise?
    - Then RIGHT_MD_A = ?, RIGHT_MD_B = ?

Moving forward

- To control moving speed,
  - Use analogWrite(EN port, value)
    - Ex) analogWrite(LEFT_MOTOR_EN, 100);
      analogWrite(RIGHT_MOTOR_EN, 100);

```
void move_forward()
{
    //Rotate counterclockwise for left motor
    digitalWrite(LEFT_MD_A, HIGH);
    digitalWrite(LEFT_MD_B, LOW);

    //Rotate clockwise for right motor
    digitalWrite(RIGHT_MD_A, LOW);
    digitalWrite(RIGHT_MD_B, HIGH);

    //Now turn left and right motors ON!
    analogWrite(LEFT_MOTOR_EN, 100);
    analogWrite(RIGHT_MOTOR_EN, 100);
}
```

Moving backward

- To make SmartCAR move backward
- Left motors
  - Should be rotated
    - counterclockwise or clockwise?
    - Then LEFT_MD_A = ?, LEFT_MD_B = ?
- Right motors
  - Should be rotated
    - counterclockwise or clockwise?
    - Then RIGHT_MD_A = ?, RIGHT_MD_B = ?

Moving backward

- To control moving speed,
  - Use analogWrite(EN port, value)
    - Ex) analogWrite(LEFT_MOTOR_EN, 100);
      analogWrite(RIGHT_MOTOR_EN, 100);

```
void move_backward()
{
    //Rotate clockwise for left motor
    digitalWrite(LEFT_MD_A, LOW);
    digitalWrite(LEFT_MD_B, HIGH);

    //Rotate counterclockwise for right motor
    digitalWrite(RIGHT_MD_A, HIGH);
    digitalWrite(RIGHT_MD_B, LOW);

    //Now turn left and right motors ON!
    analogWrite(LEFT_MOTOR_EN, 100);
    analogWrite(RIGHT_MOTOR_EN, 100);
}
```

Stop

- To stop motors,
  - Disable EN ports using digitalWrite() or analogWrite()
    - Ex) digitalWrite(LEFT_MOTOR_EN, LOW);
      digitalWrite(RIGHT_MOTOR_EN, LOW);
    - Ex) analogWrite(LEFT_MOTOR_EN, 0);
      analogWrite(RIGHT_MOTOR_EN, 0);

```
void move_stop()
{
    analogWrite(LEFT_MOTOR_EN, 0);
    analogWrite(RIGHT_MOTOR_EN, 0);
}
```

1) Move forward for 2s
2) Stop for 500ms
3) Move backward for 2s

```
#define LEFT_MD_A       22
#define LEFT_MD_B       23
#define RIGHT_MD_A      24
#define RIGHT_MD_B      25

#define LEFT_MOTOR_EN 4
#define RIGHT_MOTOR_EN          5

int init_done = false;

//The setup function is called once at startup of the sketch
void setup()
{
    // Add your initialization code here
    pinMode(LEFT_MD_A, OUTPUT);
    pinMode(LEFT_MD_B, OUTPUT);
    pinMode(RIGHT_MD_A, OUTPUT);
    pinMode(RIGHT_MD_B, OUTPUT);
    pinMode(LEFT_MOTOR_EN, OUTPUT);
    pinMode(RIGHT_MOTOR_EN, OUTPUT);

    digitalWrite(LEFT_MD_A, LOW);
    digitalWrite(LEFT_MD_B, LOW);
    digitalWrite(RIGHT_MD_A, LOW);
    digitalWrite(RIGHT_MD_B, LOW);
    digitalWrite(LEFT_MOTOR_EN, LOW);
    digitalWrite(RIGHT_MOTOR_EN, LOW);
}
```

```
void move_forward()
{
    //Rotate counterclockwise for left motor
    digitalWrite(LEFT_MD_A, HIGH);
    digitalWrite(LEFT_MD_B, LOW);

    //Rotate clockwise for right motor
    digitalWrite(RIGHT_MD_A, LOW);
    digitalWrite(RIGHT_MD_B, HIGH);

    //Now turn left and right motors ON!
    analogWrite(LEFT_MOTOR_EN, 100);
    analogWrite(RIGHT_MOTOR_EN, 100);
}
void move_backward()
{
    //Rotate clockwise for left motor
    digitalWrite(LEFT_MD_A, LOW);
    digitalWrite(LEFT_MD_B, HIGH);

    //Rotate counterclockwise for right motor
    digitalWrite(RIGHT_MD_A, HIGH);
    digitalWrite(RIGHT_MD_B, LOW);

    //Now turn left and right motors ON!
    analogWrite(LEFT_MOTOR_EN, 100);
    analogWrite(RIGHT_MOTOR_EN, 100);
}
```

1) Move forward for 2s
2) Stop for 500ms
3) Move backward for 2s

- Make sure to have a break for at least 500ms between different operations

```
void move_stop()
{
    analogWrite(LEFT_MOTOR_EN,  0);
    analogWrite(RIGHT_MOTOR_EN,  0);
}
```

```
// The loop function is called in an endless loop
void loop()
{
    //Add your repeated code here
    if (init_done == false)
    {
        move_forward();
        delay(2000);

        move_stop();
        delay(500);

        move_backward();
        delay(2000);

        init_done  =  true;
    }
    else
        move_stop();
}
```

Today

- Lab announcement
- Review Lab assignments #2-1 and #2-2
- SmartCAR Motor Control
- **Lab assignment #3**
- Course announcement

Lab Assignment #3

- 1) Move forward for 2 seconds
- 2) Stop for 0.5s
- 3) Turn right for 1.5s
- 4) Stop for 0.5s
- 5) Turn left for 1.5s
- 6) Stop for 0.5s
- 7) Move backward for 2 seconds
- 8) Stop forever!

- Upload your 1) lab3.h and 2) lab3.cpp file to Cyber Campus
- Show your result to TA or instructor

Course Announcement

- Next lecture, we will continue to study
  - Arduino Mega2560 board
  - UART & Bluetooth communication between SmartCAR and Android device

- Next lab session, we will cover
  - Bluetooth communication between SmartCAR and Nexus 7 tablet
  - Writing your own Android application for controlling SmartCAR (using App Inventor!)