CH. 4: Project Practice

[1] How to use Arduino Mega 2560

[2] Input/output interface program

- **(1) ON/OFF-type sensors and devices**
- **(2) Analog-type sensors**
- **(3) DC Motor and Servo Motor interface**

[3] PID position control of DC motors (AX-12W)

[4] 2-DOF robot arm control project (AX-12A)

[5] Walking robot project using Arduino Mega 2560

- **(1) Controller design with Arduino Mega 2560**
- **(2) Design of two-type of walking robot mechanisms**
- **(3) Prototypes**

[6] 2-DOF haptic device project

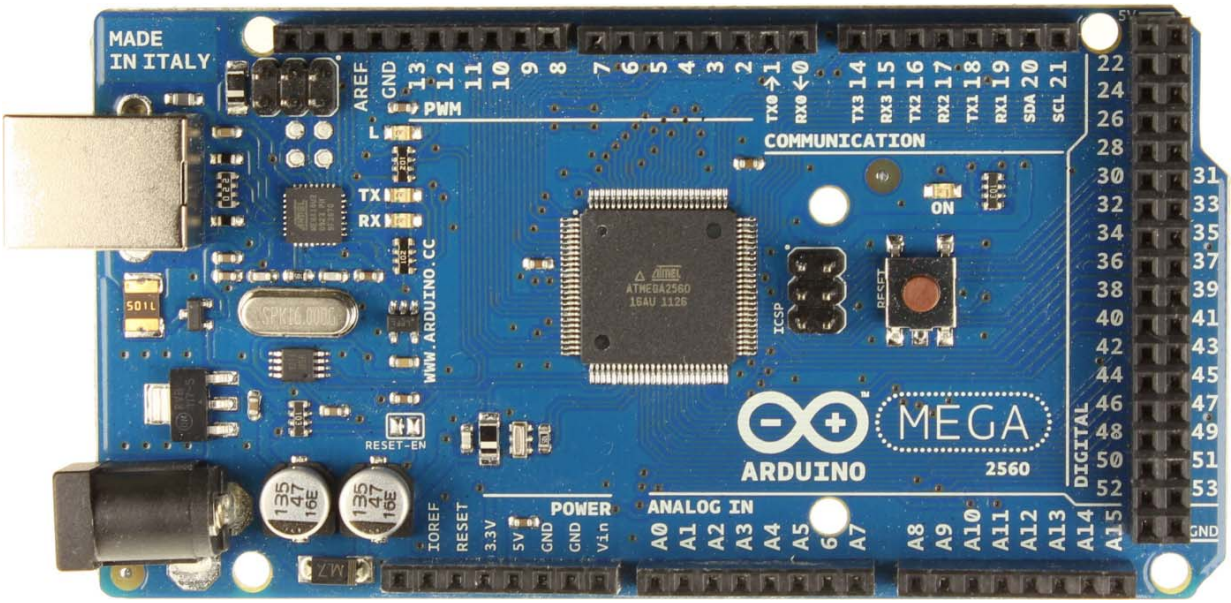
CC BY SA

CH. 4: Project Practice

[1] How to use Arduino Mega 2560

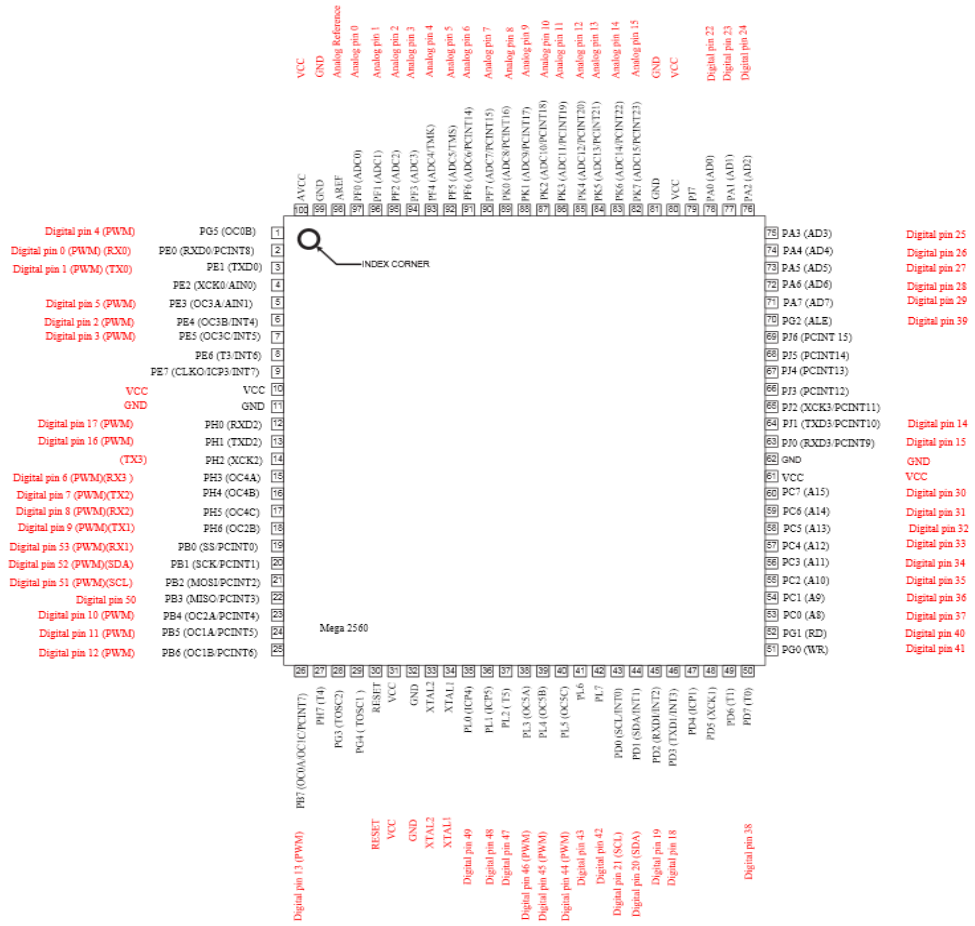
(1) Introduction

- Microcontroller: ATmega2560 (16MHz, 256KB Flash Memory), 54 DIO pins, 14 PWM Outputs, 16 Analog Inputs, 4 UARTs, USB connector
- Supported I/O: Digital Input and Output, Analog Input, Serial Receive and Transmit, Servo Read and Write
- Supported from MATLAB R2012a version.
- Related MathWorks website: http://www.mathworks.co.kr/hardware-support/arduino-simulink.html



[Arduino Mega 2560]

CC BY SA



[Pin Assignment of ATmega2560]

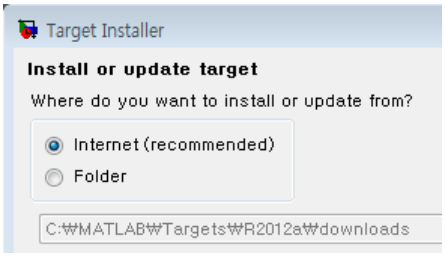
(2) Installation

- Arduino Board software

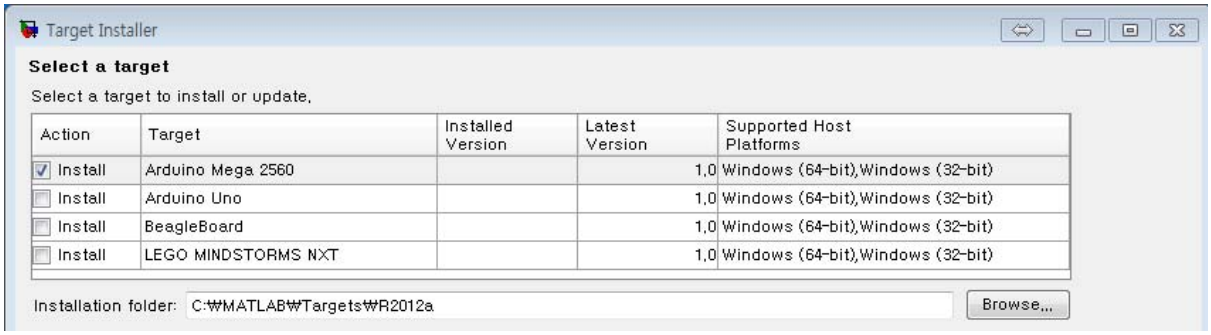
  http://arduino.cc/en/Main/Software

  Arduino-1.0.4-windows.zip

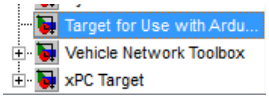
- MATLAB Command: targetinstaller



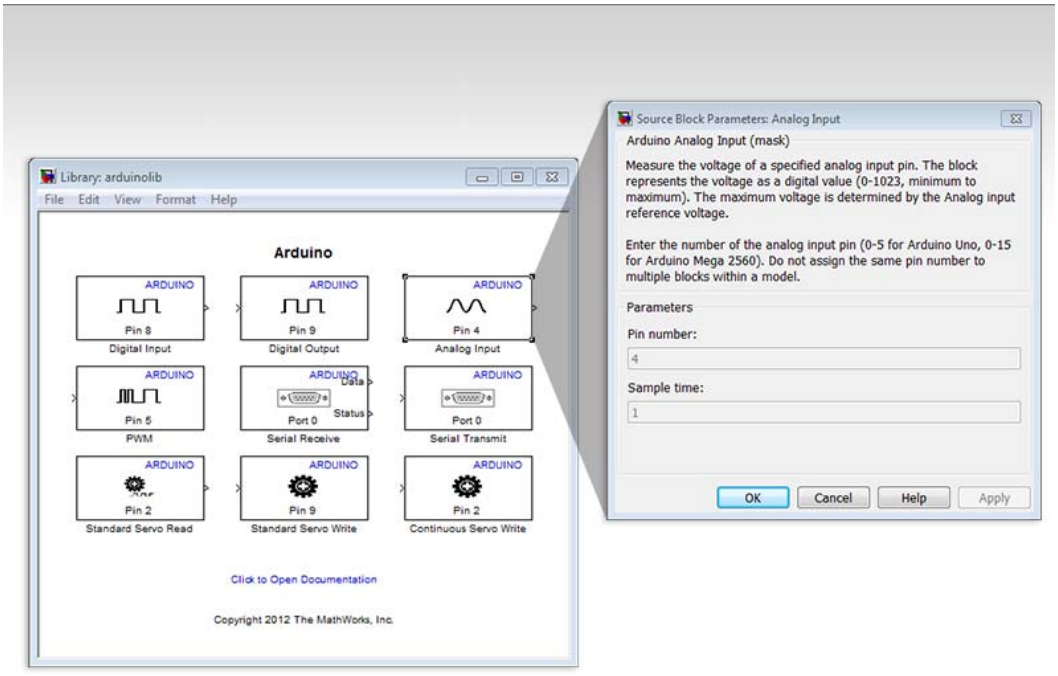
CC BY SA



- Once packages are installed, Arduino library can be found in Simulink library.

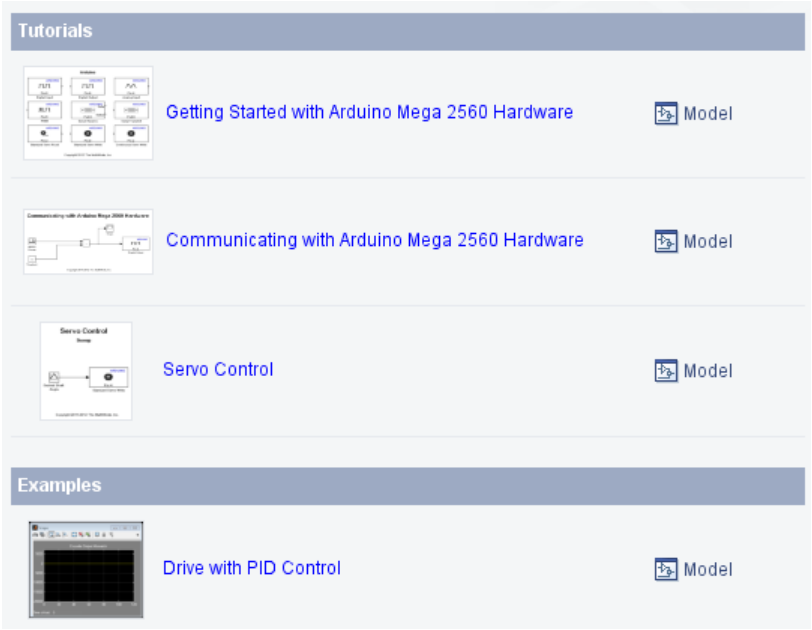


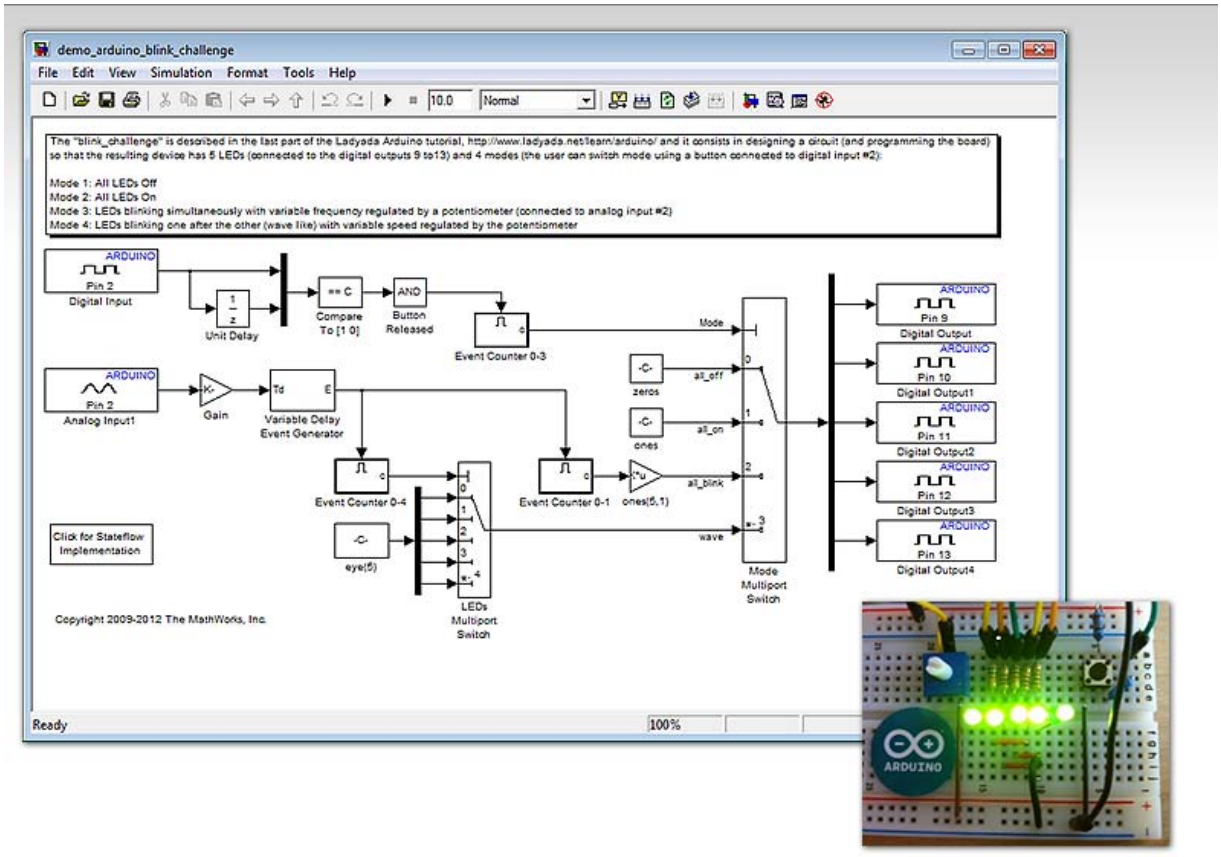
(3) I/O Drivers



CC BY SA

(4) Related demo and examples:

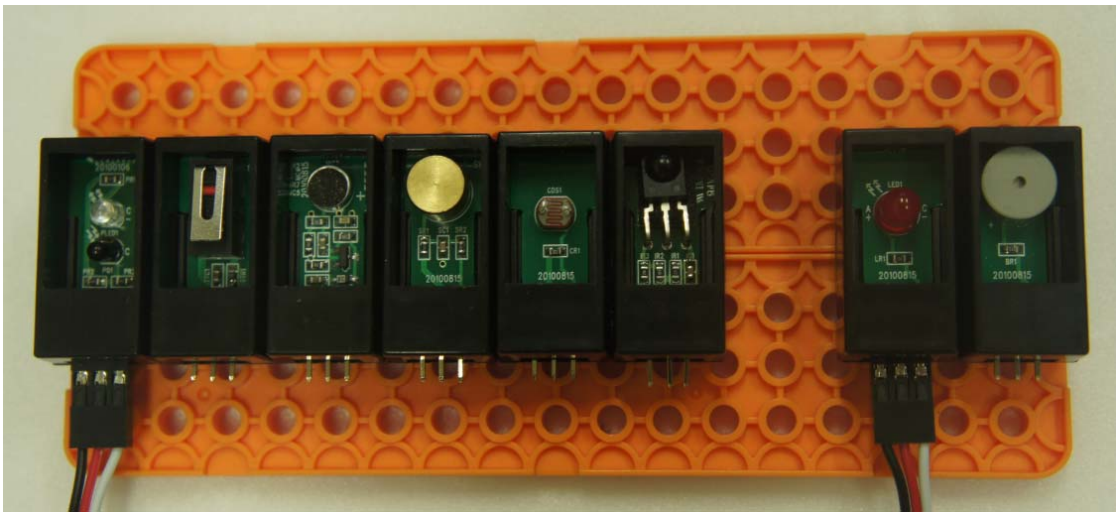




CC BY SA

[2] Input/output interface program

(1) ON/OFF-type sensors and devices



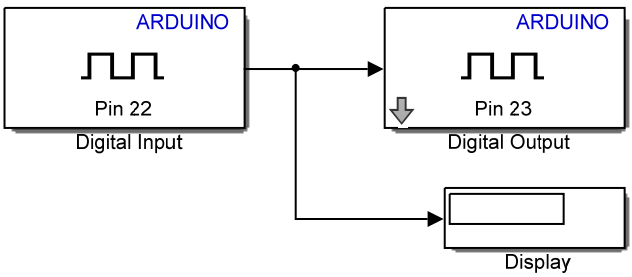
[Sensors: IR (Infrared), Switch, MIC (Microphone), Tilt, CdS, Remote sensors from the left]

[Output devices: LED, Buzzer]

- Connection: From the left, GND, VCC(+5V), Output.

Input devices	Arduino	Output devices	Arduino
GND (BLACK)	Digital GND	GND (BLACK)	Digital GND
Vcc (RED)	Digital 5V	Vcc (RED)	Digital 5V
Out (WHITE)	Digital 22pin	Out (WHITE)	Digital 23pin

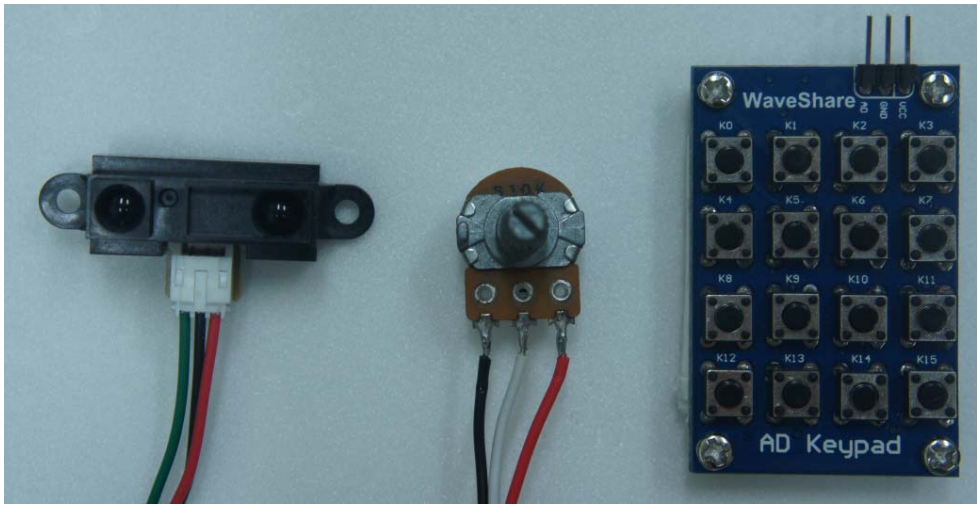
- Simulink Program: Digital Input's sample time=0.01 sec



- Execution: Tools → Run on Target Hardware → Options (Enable External Mode) → Run & Stop

CC BY SA

(2) Analog-type sensor

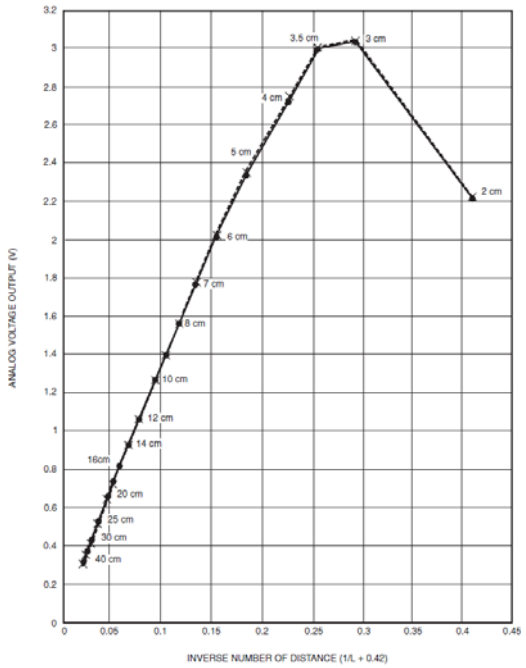


[IR sensor, Potentiometer, Analog-type Keypad]

- Connection:

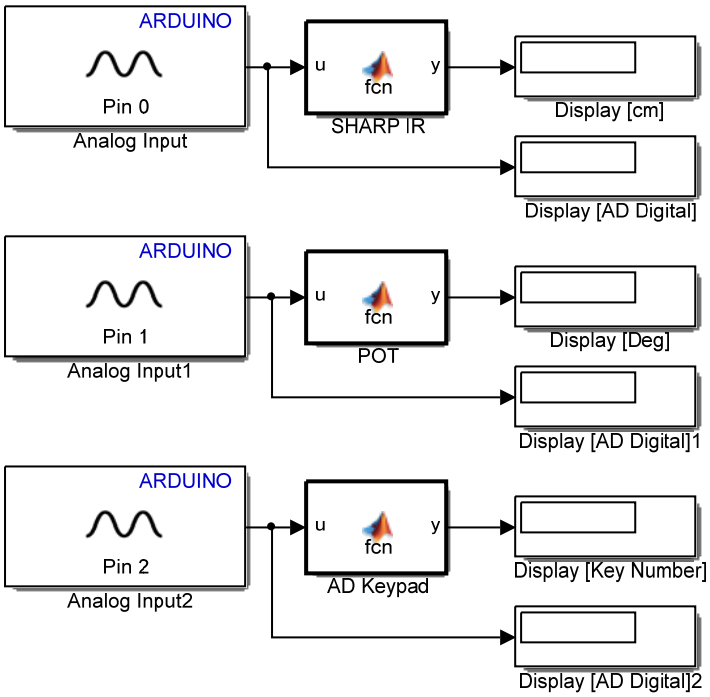
IR Sensor	Arduino	POT	Arduino	Keypad	Arduino
GND (BLACK)	Digital GND	GND (BLACK)	Digital GND	GND (BLACK)	AREF
Vcc (RED)	Digital 5V	Vcc (RED)	Digital 5V	Vcc (RED)	GND
Out (GREEN)	A0	Out (WHITE)	A1	Out (WHITE)	A2

- IR Sensor: Sharp GP2D120, Range 4~30cm



CC BY SA

- Simulink Program: Analog Input's Sample time=0.01 sec
- Arduino Mega 2560 board's ADC: 10bit (0~1023 output)



- SHARP IR:

```
function y = fcn(u)
gain=5/1024;        % 1024 for 5V %
v=gain*u;

y=0;
if v>=2.9, y=3.5; end
if v>=2.6 && v<2.9, y=4.0; end
if v>=2.2 && v<2.6, y=5.0; end
if v>=1.9 && v<2.2, y=6.0; end
if v>=1.7 && v<1.9, y=7.0; end
if v>=1.5 && v<1.7, y=8.0; end
if v>=1.35 && v<1.5, y=9.0; end
if v>=1.2 && v<1.35, y=10.0; end
if v>=1.0 && v<1.2, y=12.0; end
if v>=0.9 && v<1.0, y=14.0; end
if v>=0.8 && v<0.9, y=16.0; end
if v>=0.7 && v<0.8, y=20.0; end
if v>=0.6 && v<0.7, y=25.0; end
if v>=0.5 && v<0.6, y=30.0; end
if v<0.5, y=40.0; end
```


CC BY SA

- POT:

```
function y = fcn(u)
gain=360/1024;
y=gain*u;
```

- AD Keypad:

```
function y = fcn(u)
y=-1;
if u>=0 && u<(64-32), y=0; end
if u>=(64-32) && u<(64+32), y=1; end
if u>=(128-32) && u<(128+32), y=2; end
if u>=(192-32) && u<(192+32), y=3; end
if u>=(256-32) && u<(256+32), y=4; end
if u>=(320-32) && u<(320+32), y=5; end
if u>=(384-32) && u<(384+32), y=6; end
if u>=(448-32) && u<(448+32), y=7; end
if u>=(512-32) && u<(512+32), y=8; end
if u>=(576-32) && u<(576+32), y=9; end
if u>=(640-32) && u<(640+32), y=10; end
if u>=(704-32) && u<(704+32), y=11; end
if u>=(768-32) && u<(768+32), y=12; end
if u>=(832-32) && u<(832+32), y=13; end
if u>=(896-32) && u<(896+32), y=14; end
if u>=(960-32) && u<(960+32), y=15; end
if u<0 || u>=(960+32), y=-1; end
```

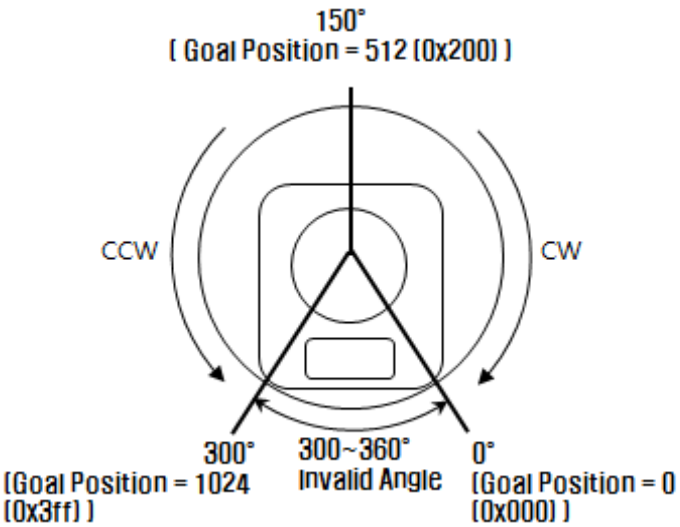

(3) DC Motor & Servo Motor interface

[DC Motor]

1) DC Motor and PWM Amplifier

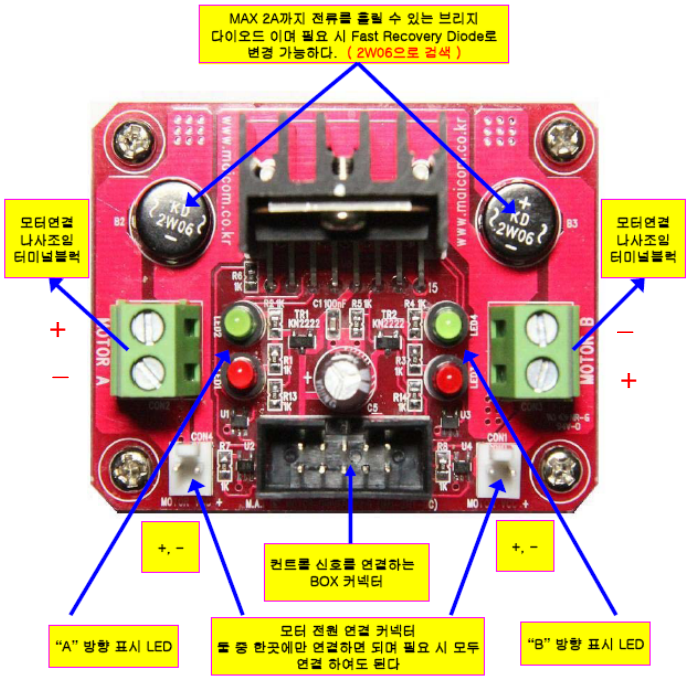
- AX-12W and AX-12A: from www.robotis.com (59rpm, 1.5Nm, 9V~12V)

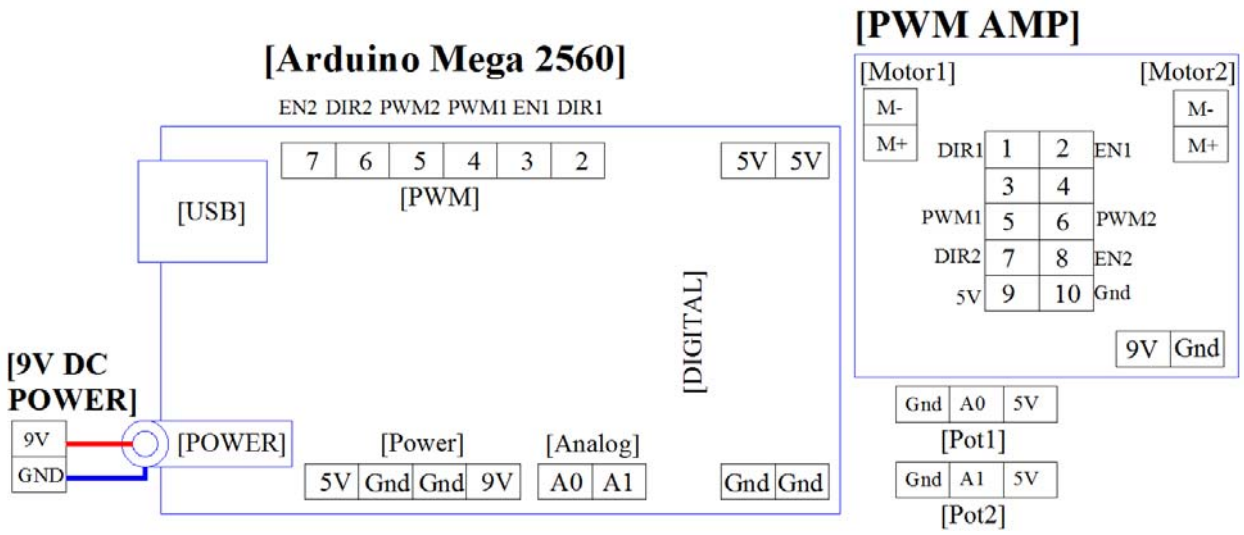
	AX-12W			AX-12A
Weight	52.9 g (1.86 oz)			54.6 g (1.88 oz)
Dimension(mm) / (inch)	32×50×40(mm) 1.25×1.97×1.57(inch)			32×50×40(mm) 1.25×1.97×1.57(inch)
Gear Ratio (material)	32 : 1 (enpla)			254 : 1 (enpla)
Network Interface	TTL			TTL
Position Sensor (Resolution)	Potentiometer (300°/1024)			Potentiometer (300°/1024)
Motor	Cored Motor			Cored Motor
Operation Voltage (V)	9.0	11.1	12.0	9.0~12.0
Stall Torque (N.m)	N/A			1.5 at 12.0V
Stall Current (A)	1.1	1.3	1.4	1.5
No Load Speed (RPM)	360	430	470	59



CC BY SA

- PWM Amplifier: MAI-2MT-DC, 2ch, 2A, 5~46V

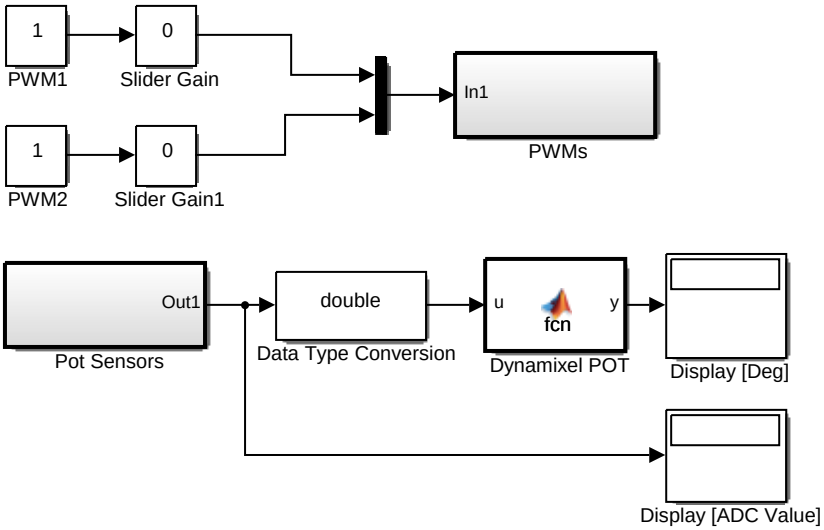




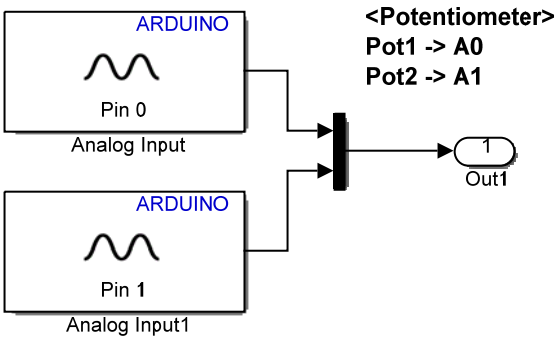
[Connection of Arduino Mega 2560 and PWM amp]

CC BY SA

- Simulink program (2ch):



- Pot Sensors:

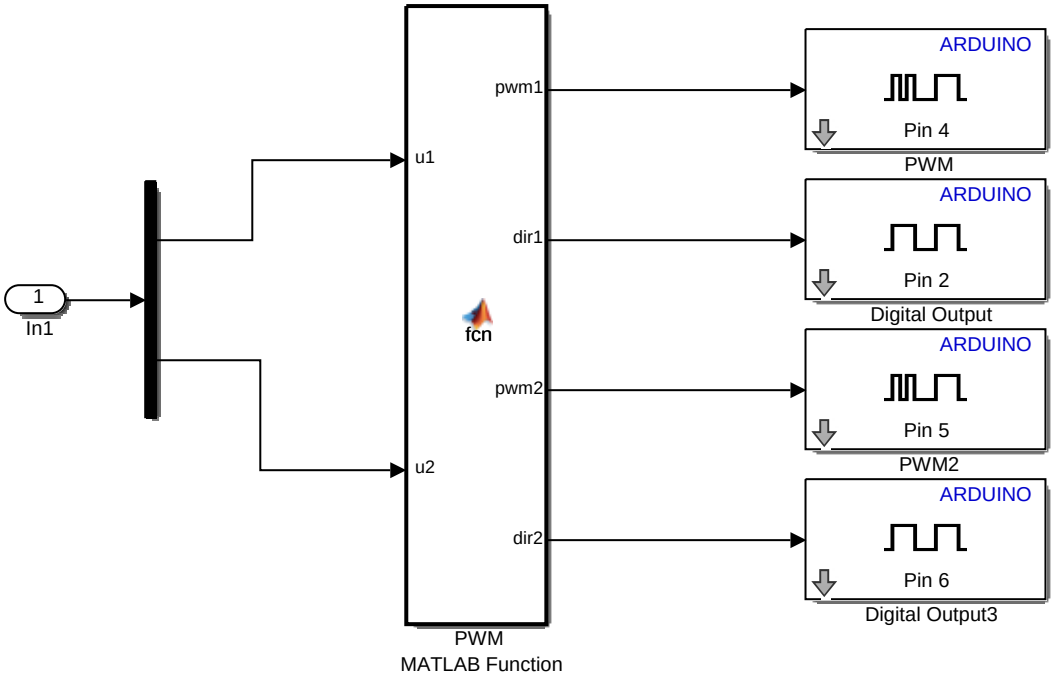


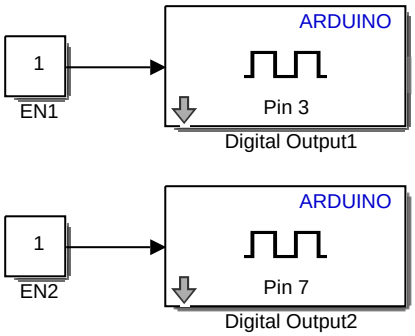
[Dynamixel POT MATLAB Function]

```
function y = fcn(u)
MAX_CH=2;
y=zeros(MAX_CH,1);
for ch=1:MAX_CH
      y(ch,1)=(300/1024)*(u(ch,1)-204.8);
end
```


CC BY SA

- PWMs Subsystem:





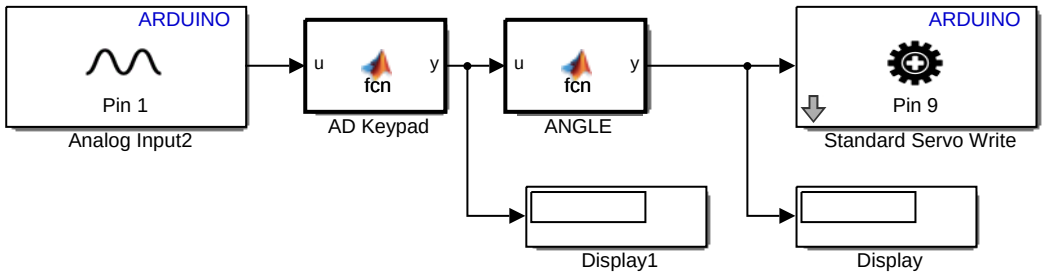
[Pin Assignments]
<DC Motor Driver>
DIR1 (1) -> Pin2
EN1(2) -> Pin3
PWM1(5) -> Pin4
PWM2(6) -> Pin5
DIR2 (7) -> Pin6
EN2(8) -> Pin7

[PWM MATLAB Function]

```
function [pwm1, dir1, pwm2, dir2] = fcn(u1, u2)
if u1>=0,dir1=1;
else dir1=0; end
if u2>=0,dir2=1;
else dir2=0; end
pwm1=abs(u1);
pwm2=abs(u2);
% Saturation %
if pwm1>=255, pwm1=255; end
if pwm2>=255, pwm2=255; end
```

2) Servo Motor

- Simulink program:



- ANGLE MATLAB Function:

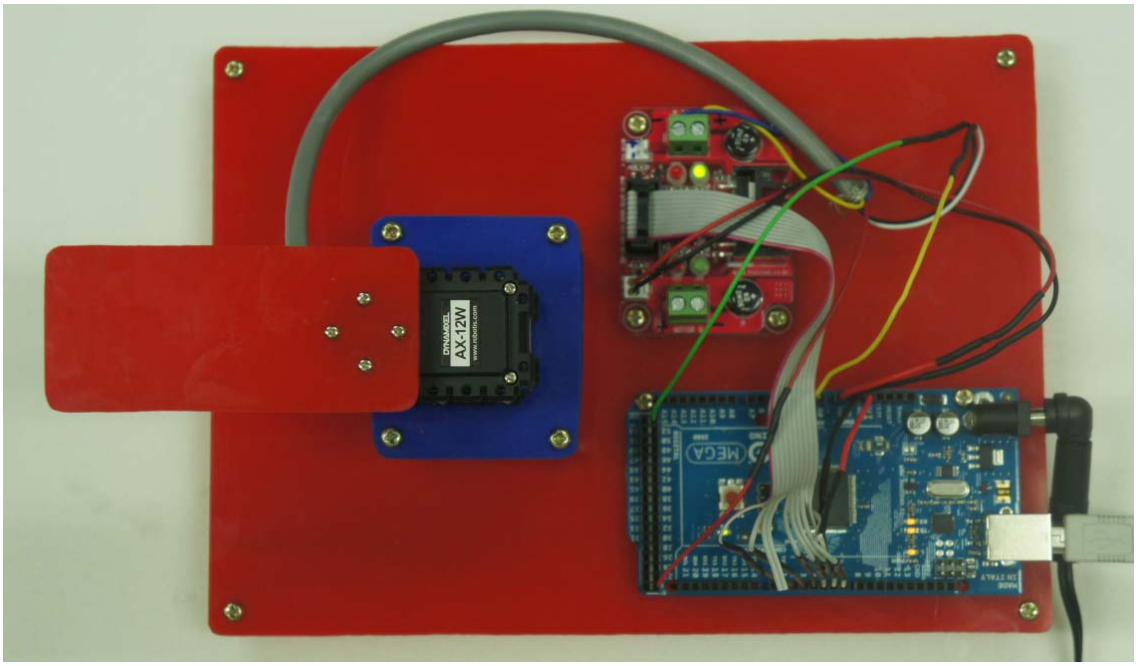
```
function y = fcn(u)
persistent ang
dth=1;
if isempty(ang), ang=90; end
if u==0, ang=ang-dth; end
if u==4, ang=ang+dth; end
if ang<0, ang=0; end
if ang>180, ang=180; end
y=ang;
```

- Operation by keypad: K0 → + direction up to 180 deg, K4 → - direction upto 0 deg
- Connection:

Servo Motor	Arduino	Keypad	Arduino
GND (BLACK)	Digital GND	GND (BLACK)	AREF
Vcc (RED)	Digital 5V	Vcc (RED)	GND
Data (YELLOW)	PWM 9Pin	Out (WHITE)	A2

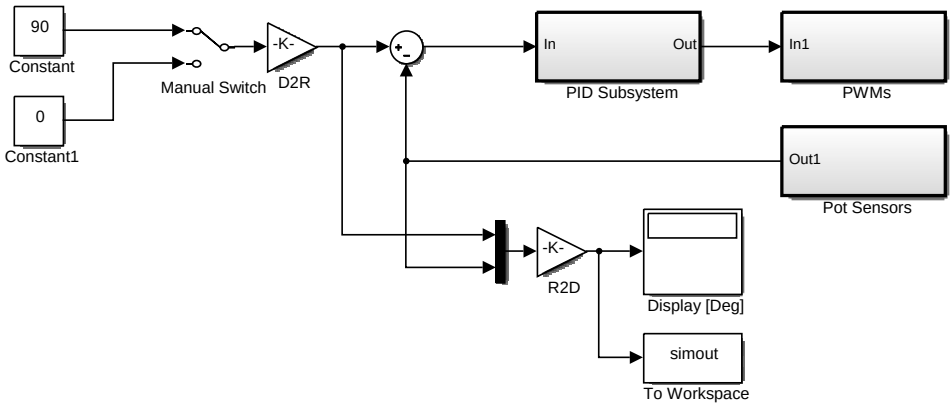
[3] PID position control of DC motors (AX-12W)

- In this experiment, AX-12A(32:1 gear ratio) is used instead of AX-12W(254:1 gear ratio.



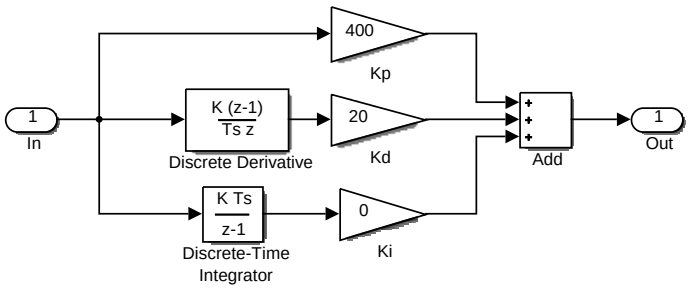
[Test bed]

- Simulink Program:



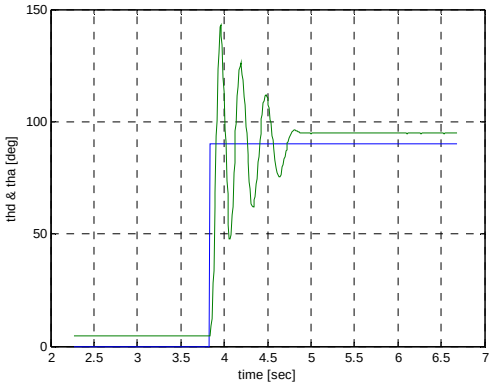
CC BY SA

- PID Subsystem:

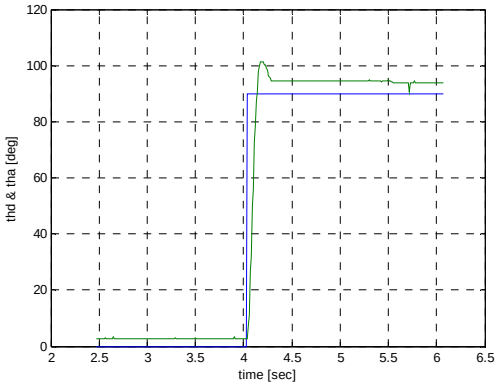


Kp=400, Kd=20

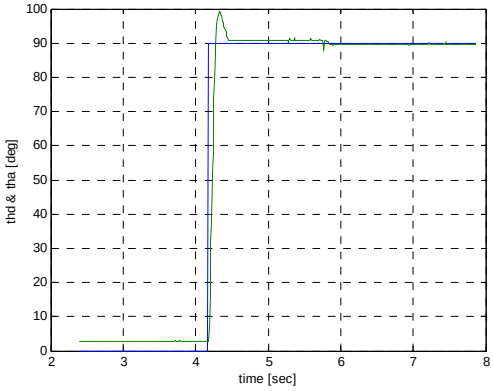
- Experiment results (blue line: desired value, green line: actual value)



[Kp=300, Kd=0, Ki=0]

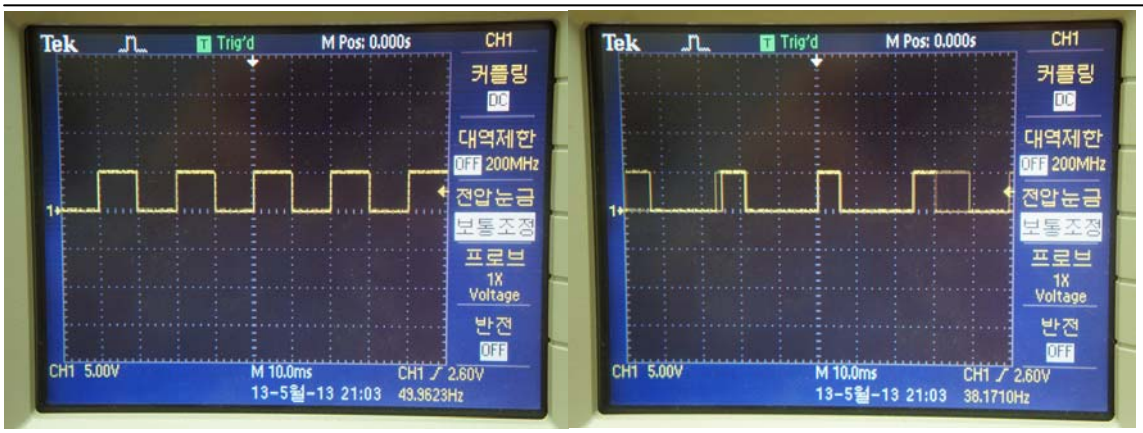


[Kp=300, Kd=12, Ki=0]



[Kp=300, Kd=12, Ki=100]

CC BY SA

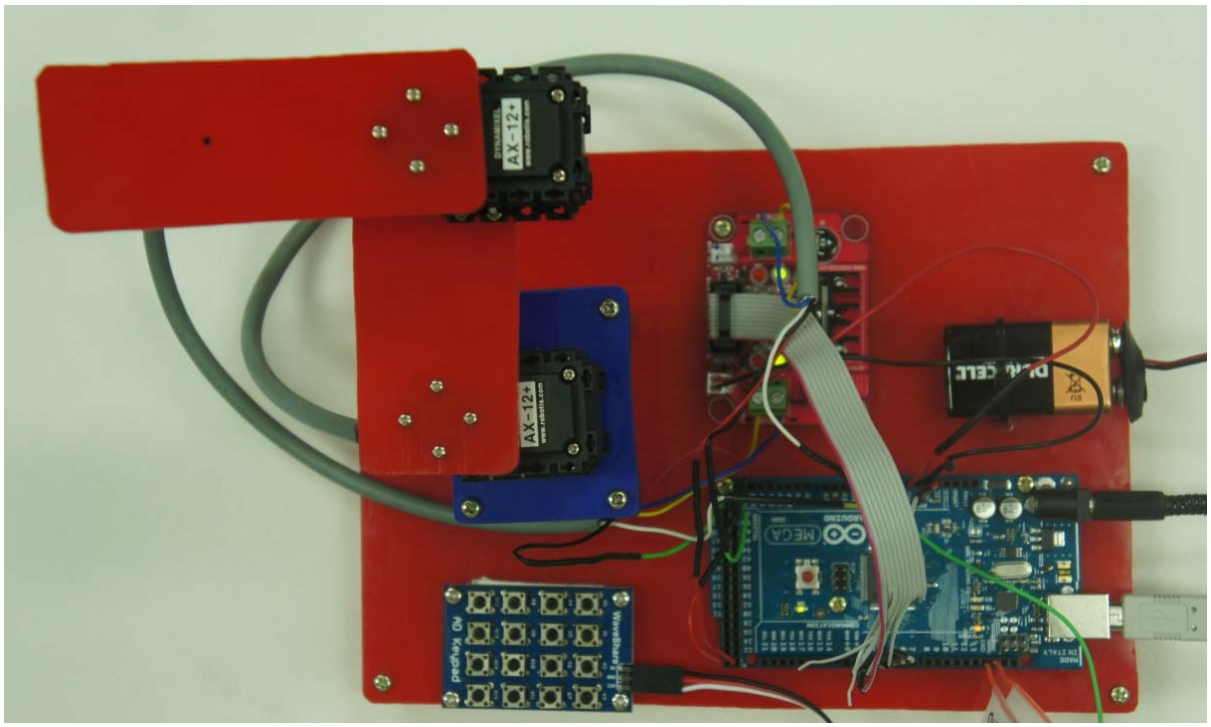


[No external mode] [External mode]

- When serial communication is used (external mode), there is some delay and loop timing is not exact. However, loop timing (for 10 msec) is fairly exact with no serial communication.

CC BY SA

[4] 2-DOF robot arm control project (AX-12A)



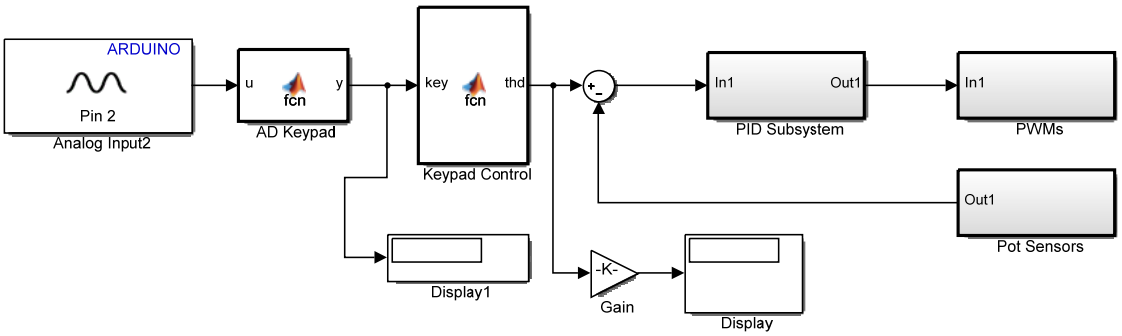
[Testbed]

- Key assignment and operation:

  Forward kinematics mode: K0→+th1, K4→-th1, K1→+th2, K5→-th2

  Inverse kinematics mode: K2→+x, K6→-x, K3→+y, K7→-y

- Simulink program:



CC BY SA

- Keypad Control MATLAB Function:

```
function thd = fcn(key)
persistent x_p th_p
D2R=pi/180;
dpos=0.5; dth=0.5*D2R;
a1= 70; a2=70;
param=[0,a1,0; 0,a2,0];
if isempty(x_p), x_p=[a1,a2]'; end
if isempty(th_p), th_p=[0,90*D2R]'; end
th=th_p; x=x_p;

switch key
    case 0, th(1,1)=th_p(1,1)+dth; index=1;          % +th1 Movement %
    case 4, th(1,1)=th_p(1,1)-dth; index=1;          % -th1 Movement %
    case 1, th(2,1)=th_p(2,1)+dth; index=1;          % +th2 Movement %
    case 5, th(2,1)=th_p(2,1)-dth; index=1;          % -th2 Movement %

    case 2, x(1,1)=x_p(1,1)+dpos; index=2;           % +X Movement %
    case 6, x(1,1)=x_p(1,1)-dpos; index=2;           % -X Movement %
    case 3, x(2,1)=x_p(2,1)+dpos; index=2;           % +Y Movement %
    case 7, x(2,1)=x_p(2,1)-dpos; index=2;           % -Y Movement %

    case 15, x=[a1,a2]'; th=[0,90*D2R]'; index=1;    % Original Pos & Orien %
    otherwise
        th=th_p; index=1;
end

switch index
    case 1, % Forward Kinematics %
        if (th(1,1)<-30*D2R || th(1,1)>=D2R*210) || (th(2,1)<-30*D2R || th(2,1)>=D2R*210),
            th=th_p;
        end
        T02=forward_kinematics(th, param);
        x=T02(1:2,4);
    case 2, % Inverse Kinematics %
        if (x(1,1)<-(a1+a2) || x(1,1)>+(a1+a2)) || (x(2,1)<-0.5*a1 || x(2,1)>+(a1+a2))
            x=x_p;
        end
        [th_c, w_index]=inverse_kinematics(x, param);
        if w_index==1, th=th_c; end
end
thd=th;
% Back substitution %
th_p=th; x_p=x;
```

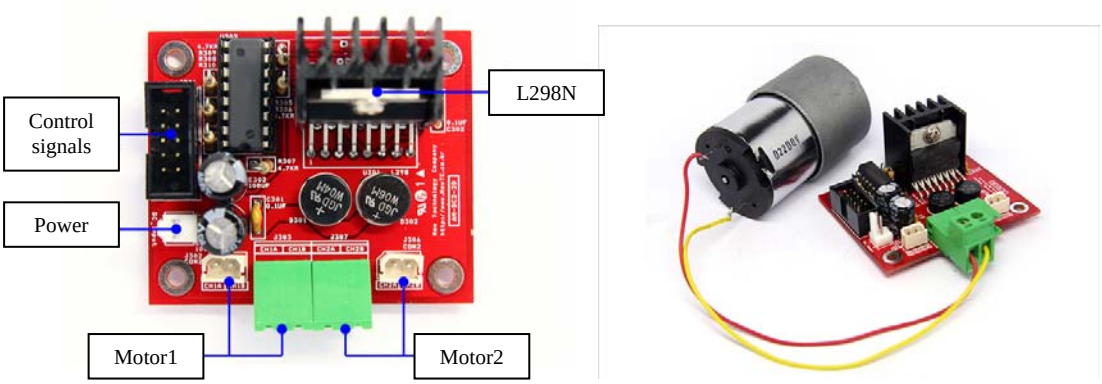

CC BY SA

[5] Walking robot project using Arduino Mega 2560

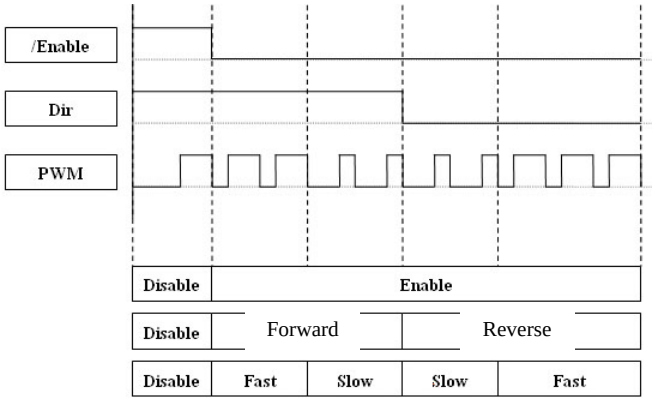
(1) Arduino board controller

- One 2-ch PWM amplifier, three IR sensors
- DC Motor Amplifier (AM-DC2-2D):

  2ch, Max. current: 2A, Operation voltage: 5~45V



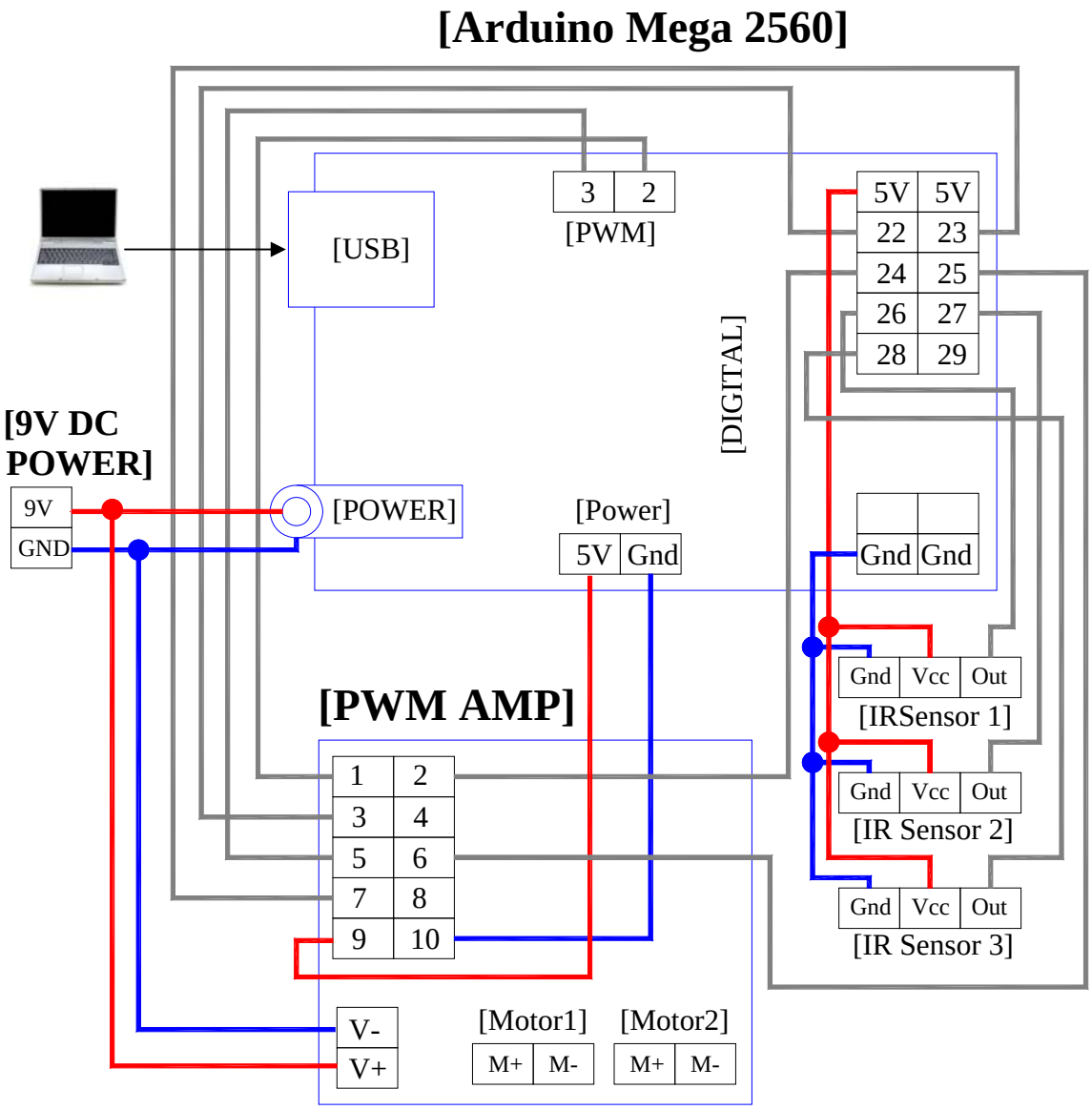
[Control Connector Pin Number]



[Control Logic]

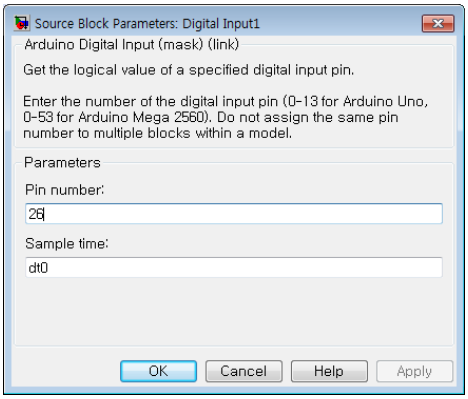
- IR Sensor: (GND (Black), RED (Vcc), WHITE (Out))

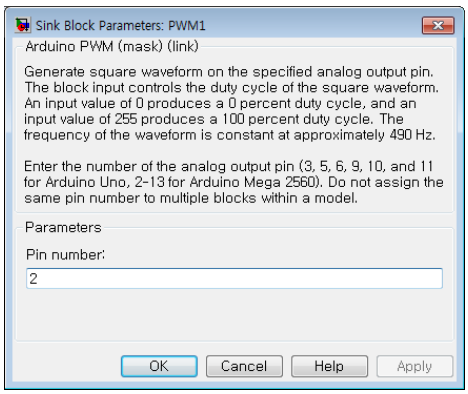
CC BY SA



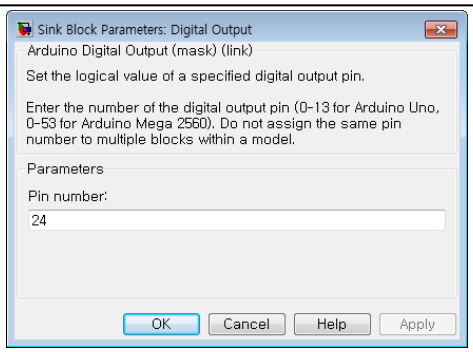
[Arduino controller connection]

- Control program:

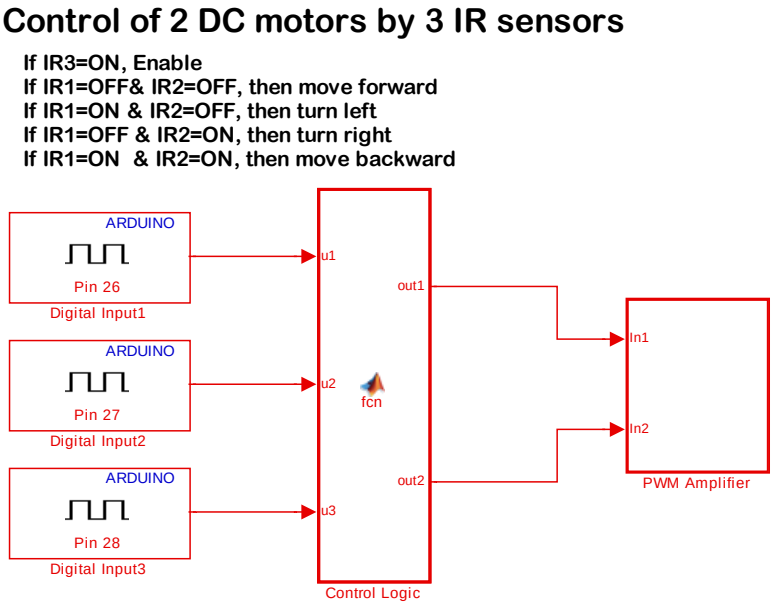




CC BY SA



[Arduino I/O setting]



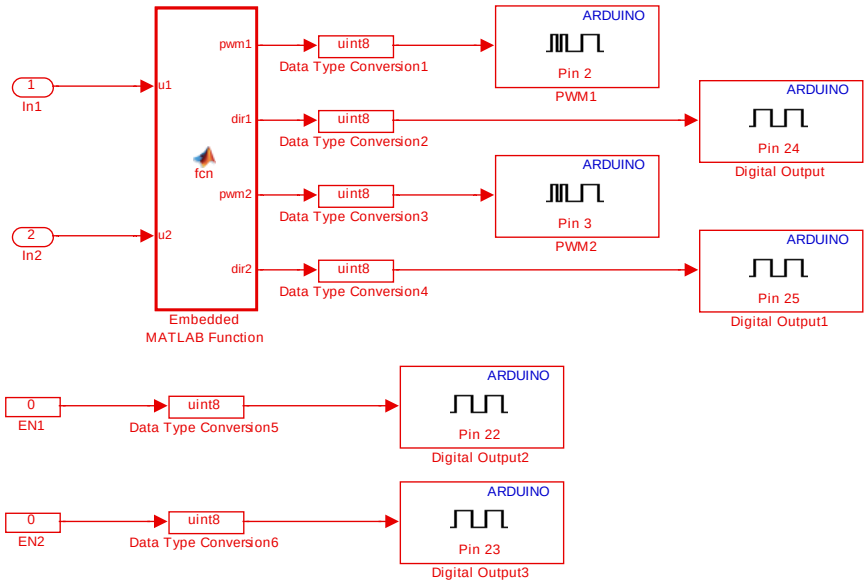
[Main control program]

<Control Logic>

```
function [out1, out2] = fcn(u1, u2, u3)

% When IR sensor is blocked (or reflected), u=0
% When IR sensor is not blocked (or not reflected), u=1
out1=0; out2=0;
%sp1=250; sp2=-250;
sp1=-190; sp2=170;          % 0~255 %
if u3==0,
      if u1==1 && u2==1,    out1=sp1; out2=sp2; end       % Forward %
      if u1==0 && u2==1,    out1=0; out2=sp2; end        % Left %
      if u1==1 && u2==0,    out1=sp1; out2=0; end        % Right %
      if u1==0 && u2==0,    out1=-sp1; out2=-sp2; end       % Backward %
else
      out1=0; out2=0;
end
```


CC BY SA

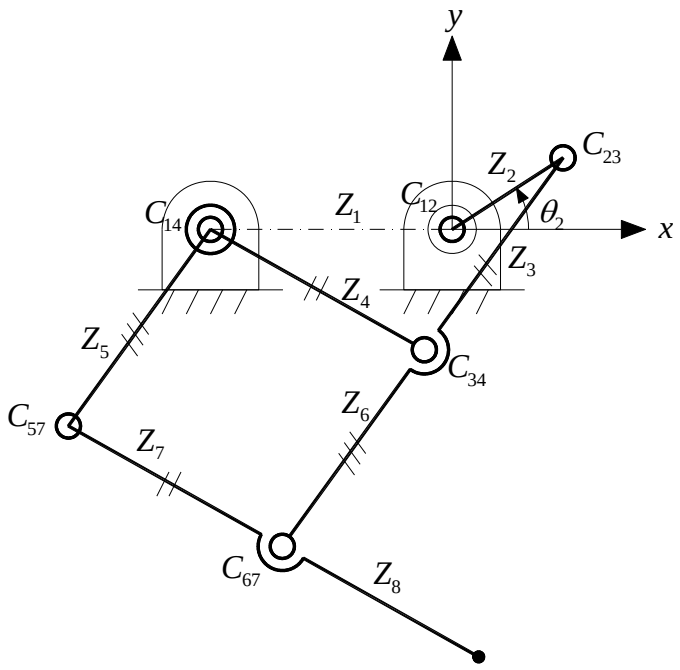


[PWM Amplifier Subsystem]

CC BY SA

(2) Walking Robot Mechanism

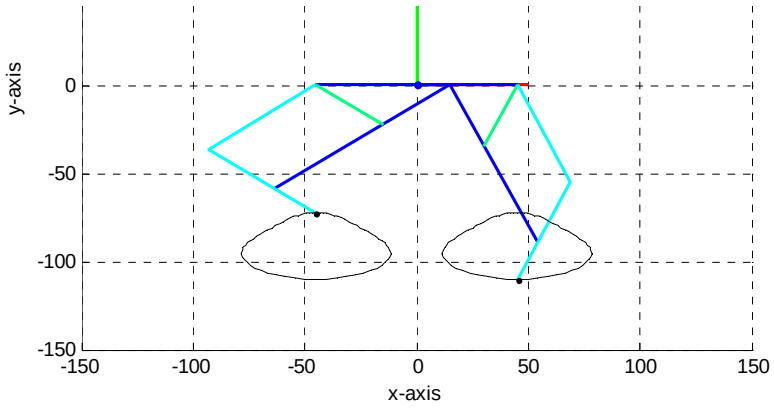
1) Walking Robot with 6-bar Legs



[6-bar Mechanism]

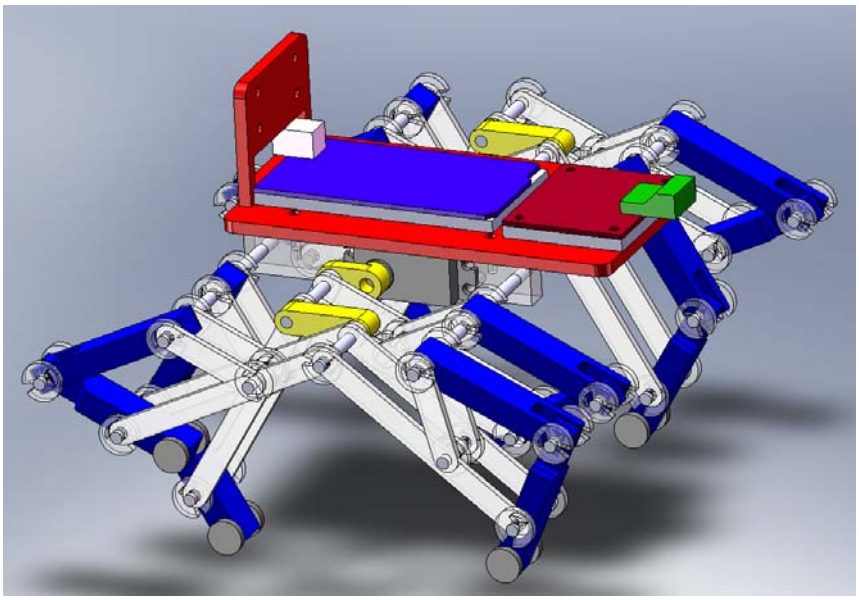
- Optimal Link Lengths:

Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8
45	15	37.5	37.5	60	60	37.5	22.5



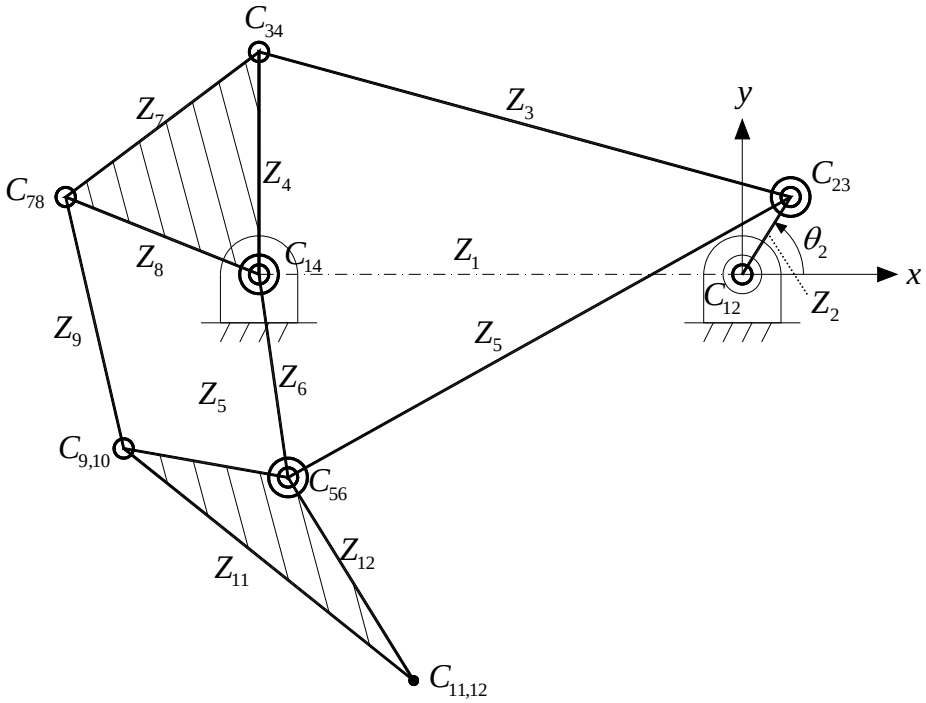
[Kinematic Simulation]

CC BY SA



[Walking Robot with 6-bar legs]

2) Walking Robot with 8-bar Legs



[8-bar Mechanism]

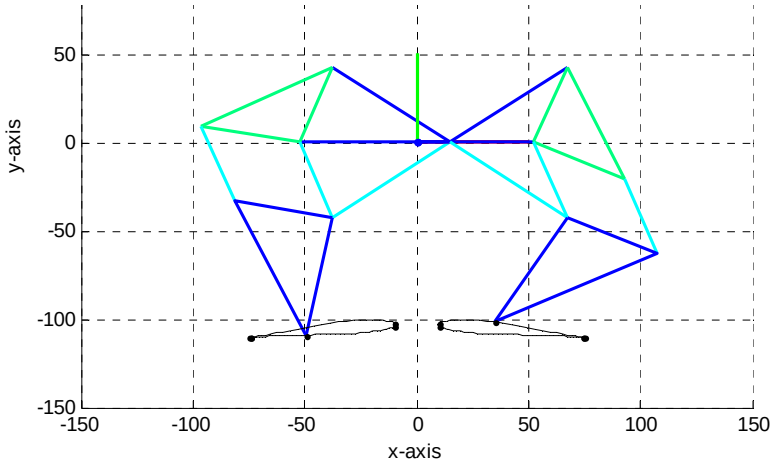
CC BY SA

- Calculation of DOF (binary links (6), Ternary links (2), single joint (4), double joint (3)):

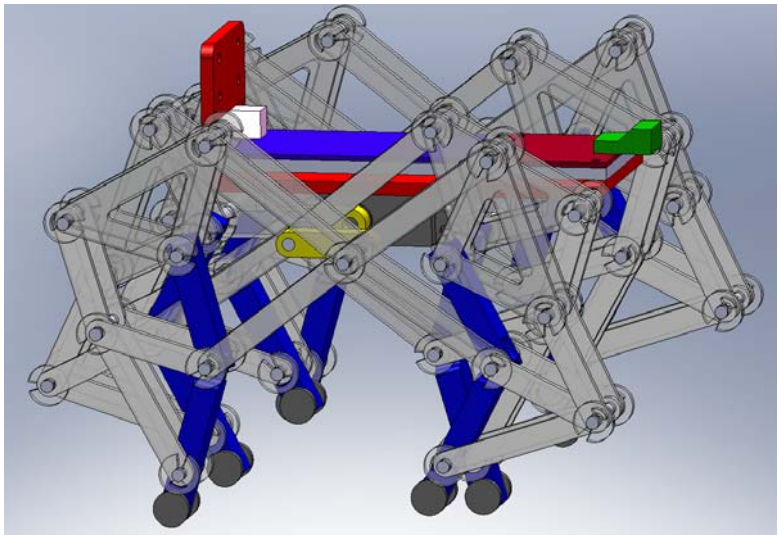
$$\begin{aligned}\text{DOF} &= \lambda(n-j-1)+\sum_{i=1}^{j} f_i \\ &= 3(8-10-1)+(4+2\times 3) \\ &= 1\end{aligned}$$

- Optimal Link Lengths:

Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}	Z_{11}	Z_{12}
90	28	128	83	128	83	115	83	83	83	166	144



[Kinematic Simulation]



[Walking Robot with 8-bar legs]

CC BY SA

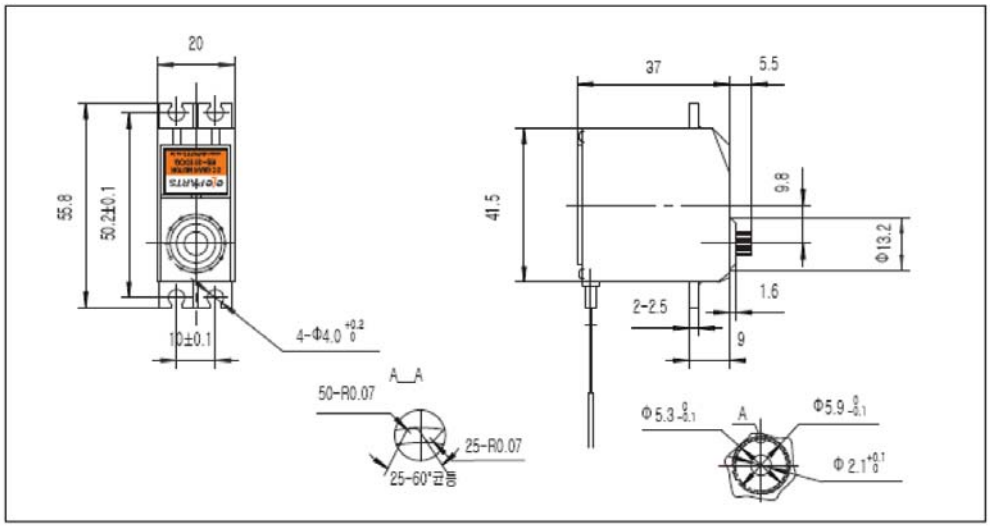
(3) Prototypes

- Geared DC Motor (Eleparts, ES-311DC): 60 RPM, 3.5kgf.cm at DC 6V

■ SPECIFICATION

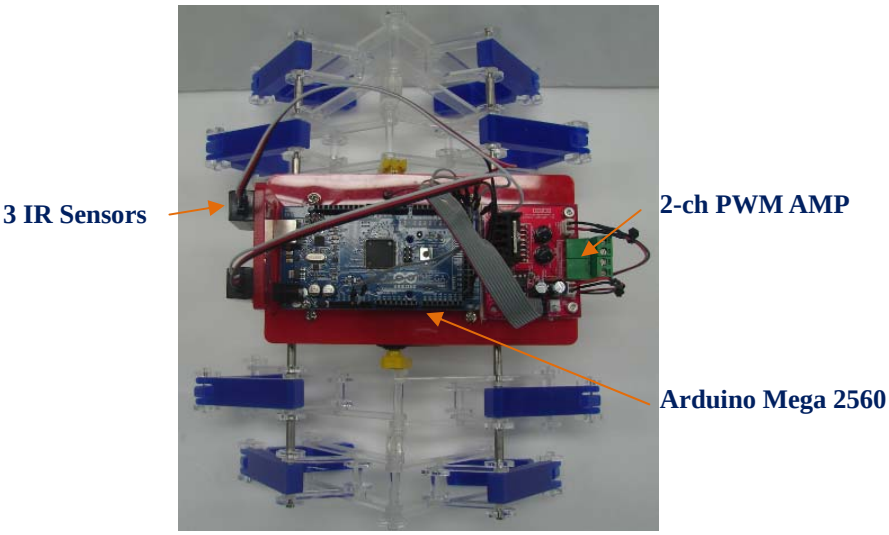
Motor Type	3 Bole Ferrite
Bearing Type	Nylon Bushing
Torque 4.8/6.0v	3.0 / 3.5 kg.
Speed 4.8/6.0v	0.19 / 0.15 second
Size	41 x 20 x 37mm
Weight	48.5 g.

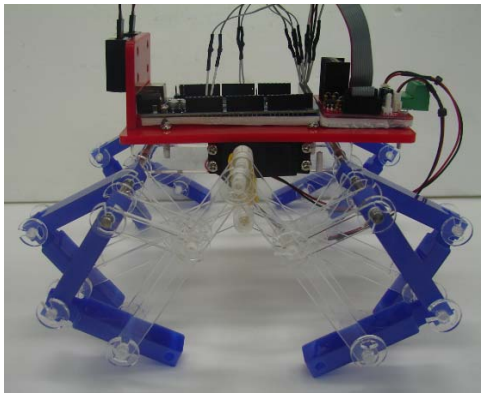
Dimension

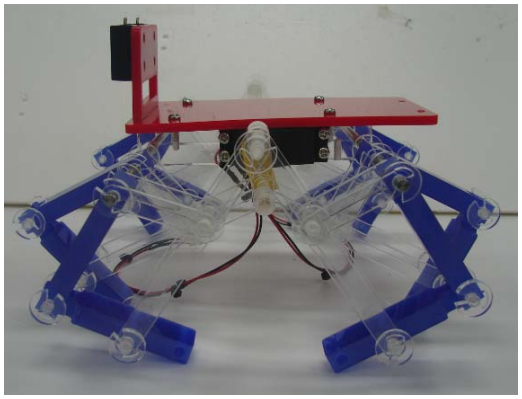


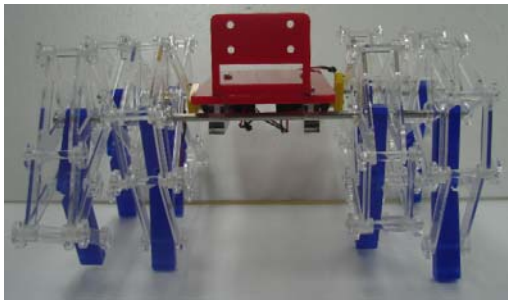
CC BY SA

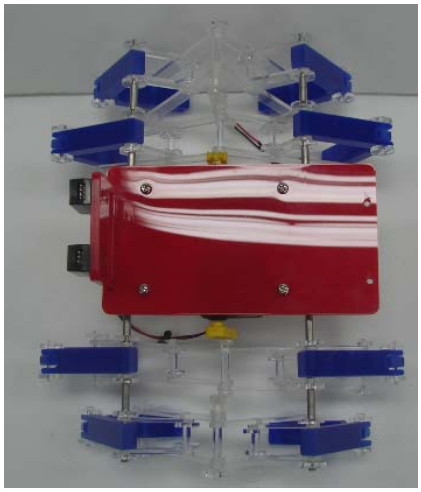
1) Walking Robot with 6-bar Legs:





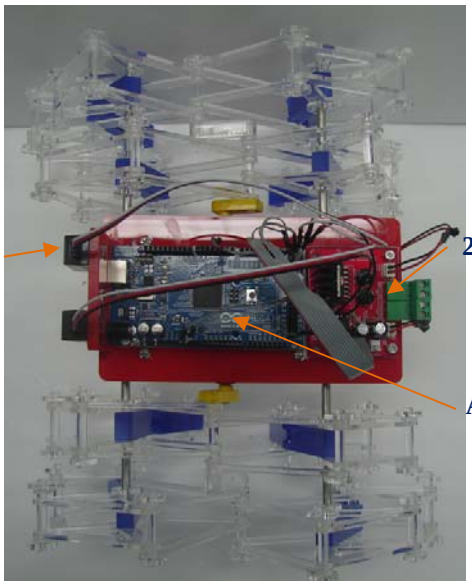


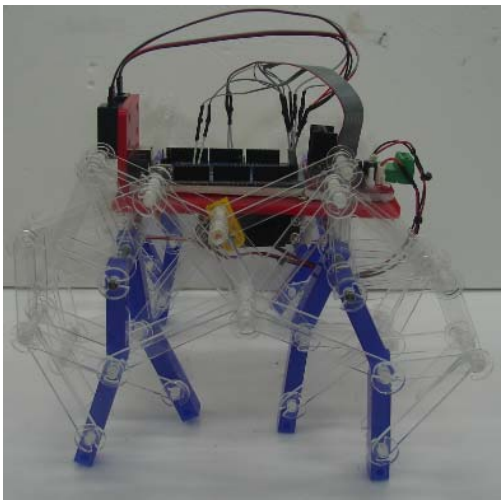


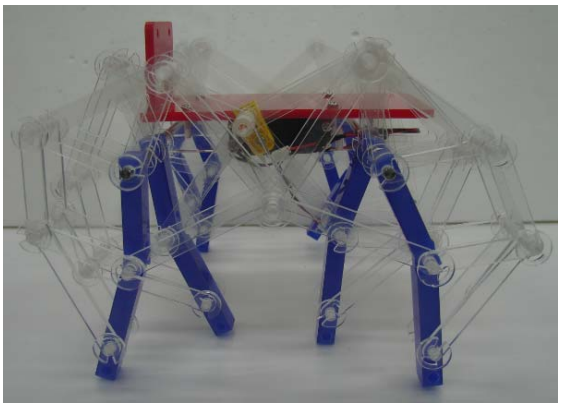


CC BY SA

2) Walking Robot with 8-bar Legs:







CC BY SA

[6] 2-DOF haptic device project

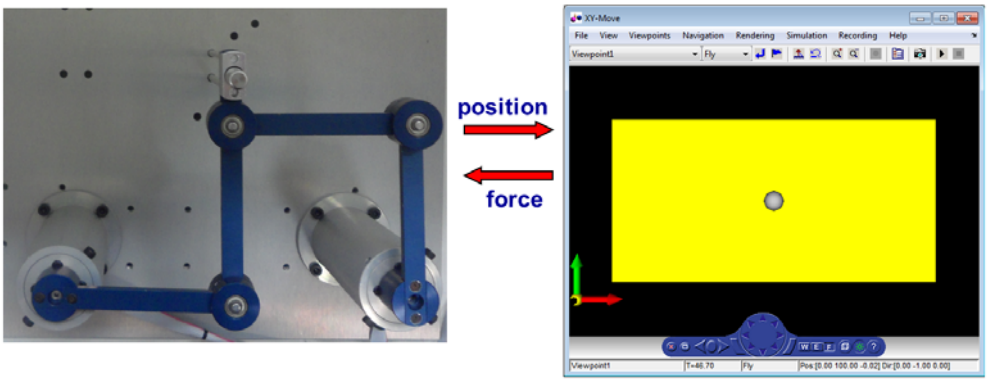
- The 5-bar (Type II) robot is used for haptic interface.
- From the two encoder angles, the end-effector position of the 5-bar robot is calculated by the forward kinematics.
- The ball in the virtual environment is moved by the input position.
- If the ball contacts the wall, then the reaction force is calculated by the program. The force information is sent to the 5-bar robot and the 5-bar robot reflects the force to an operator.

◆ **2-DOF Parallel-type Haptic Device**

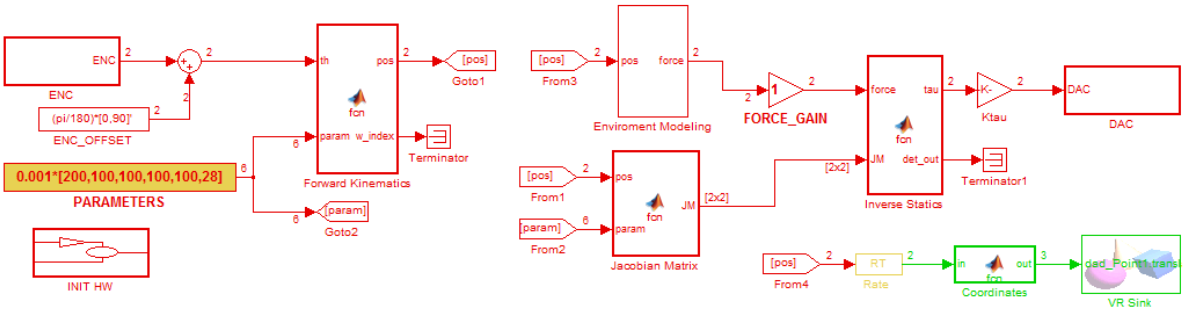
✓ Forward kinematics module

✓ Force-Reflecting module

✓ VRML graphic module



[Haptic Device] [Virtual Environment]



CC BY SA