ขั้นตอนการติดตั้งโปรแกรม AVR Studio 4

CD ที่แถมมาให้กับชุดหุ่นยนต์ AVR จะมีชุดโปรแกรม avr studio V4 มาให้แล้ว และสามารถทำการติดตั้งตาม ขั้นตอนดังนี้



Folder / avr studio V4

ภายใน CD จะมี Folder / avr studio V4 ซึ่งภายใน Folder ดังกล่าวจะมีโปรแกรมอยู่ 3 ชุดคือ aStudio4b528 / AVRStudio4.13SP2 และ WinAVR-20070525-install





จะมีชุคโปรแกรมอยู่ 3 ชุค

- ในการติดตั้งโปรแกรมจะต้องทำการติดตั้งแบบเรียงคำดับดังนี้
- 1. ทำการติดตั้ง aStudio4b528 ก่อน
- 2. ตามด้วยการติดตั้ง AVRStudio4.13SP2
- 3. และการติดตั้ง WinAVR-20070525-install

ขั้นตอนการติดตั้ง aStudio4b528



1. เลือกที่ aStudio4b528



2. โปรแกรมอยู่ระหว่างการติดตั้ง







4. เลือกที่ I accept the terms of the license agreement และ เลือก Next
 5.เลือก Next (หากต้องการเปลี่ยนที่อยู่ที่เก็บของโปรแกรมก็สามารถเลือกที่ Change ได้)





6. เลือก Next

7.เลือก Install เพื่อทำการติดตั้งโปรแกรม



8. โปรแกรมอยู่ระหว่างการติดตั้ง และรอจนกว่าจะเสร็จสมบูรณ์
 9.เมื่อ โปรแกรมทำการติดตั้งเสร็จแล้วให้เลือก Finish

จากนั้นให้ทำการติดตั้ง AVRStudio4.13SP2 ต่อเนื่องไปเลย ทำการติดตั้ง AVRStudio4.13SP2



AVRStudio4.13SP2 Setup.exe Macrovision Corporation

1. เลือกที่ AVRStudio4.13SP2



2.เข้าสู่การติดตั้งโปรแกรม











5. โปรแกรมกำลังติดตั้งรองนกว่างะเสร็ง



6. เมื่อเสร็จสิ้นการติดตั้งให้เถือก Finish

จากนั้นให้ทำการติดตั้ง WinAVR-20070525-install ต่อไปเลยครับ <u>ทำการติดตั้ง WinAVR-20070525-install</u>



1. เลือก WinAVR-20070525-install



2. เลือก OK



3.เลือก Next



🗟 WinAVR 20070525 Setup	
Choose Install Location Choose the folder in which to install WinAVR 20070525.	
Setup will install WinAVR 20070525 in the following folder. To install in a different folder, o Browse and select another folder. Click Next to continue.	ilick.
Destination Folder EstwinAts250000588 Browse	ו
Space required: 90.1MB Space available: 3.7GB	_
< Back Next > Can	cel

- 4. เลือก I Agree
- 5. เลือก Next (ถ้าต้องการเปลี่ยนที่ติดตั้งโปรแกรมก็สามารถทำได้โดยเลือกที่ Browse)





- 6. เลือก Install เพื่อทำการติดตั้ง โปรแกรม
- 7. โปรแกรมอยู่ระหว่างการติดตั้ง



8. เมื่อเสร็จสิ้นการติดตั้งแล้วให้เลือก Finish

เมื่อเสร็จสิ้นการติดตั้งทั้ง 3 โปรแกรมแล้วให้ตรวจสอบว่ามีโปรแกรม avr studio V4 พร้อมใช้งานหรือไม่



ขั้นตอนต่อไปจะเป็นการเขียนโปรแกรมเบื้องต้นและการทดสอบหุ่นยนต์ ครับ ซึ่งจะอยู่ในกู่มือการเขียนโปรแกรมและ ทดสอบหุ่นยนต์ ครับ

ขั้นตอนการเขียนโปแกรม

ขั้นตอนนี้จะเป็นการเรียนรู้การใช้งานโปรแกรม AVR Studio 4 การเขียนโปรแกรมทคสอบหุ่นยนต์เบื้องต้นกันครับ ลำดับแรกให้ทำการสร้าง Folder ที่ใช้ในการเก็บ code กันก่อนครับ ซึ่งในที่นี้จะทำการสร้าง Folder ชื่อ robodkit



Folder ที่สร้างขึ้นมาเพื่อเก็บ code program

<u>การใช้งาน AVR Studio 4</u>

1. เปิดโปรแกรมขึ้นมาก่อน โดยเข้าไปที่ start / All Program / Atmel AVR Tools / AVR Studio4



การเปิดโปรแกรม



กำลังเข้าสู่โปรแกรม

Trace Disabled *		- 7 7 ····	0000		10000
			I/O Vew	- 11	1000
	Welcome to AVP Stu	white at	and the second s	in the second	
	Welcoule to XVK Sto	1010 4			
		New Project	Open		
	150				
	10.00	Recent projects	Modified		
	8 0				
				-	_
				dress Value Bits	
	Ver 4.13.571 🕅 Show	dialog at startup			
		scBack Next 2 x	French Cancel Hel	p	
essage					
essage Loaded plugin STR 500					
essage Loaded plugin STX500					
lessage Loaded plugin STX500					
lessage Loaded plugin STK500					
tessage Loaded plugin STX500					
kessage Loaded plugin STR500					
lessage Loaded plugin STR500					

หน้าต่างโปรแกรม

2. ให้ทำการสร้าง Project ใหม่ โดยการเลือก New Project

Welcome to AVR Stu	dio 4		
15	New Project	Open	
A Str	Recent projects	-	Modified
50			
SS			
2			
Ver 4.13.571 🔽 Show	dialog at startup	1	
	KK Back	Next>> Finish	Cancel Help

เถือกที่ New Project

3.เลือกที่ AVR GCC

Project type:	Project name:	
	Assembler Create initial file Initial file:	Create folder
3		.0
Location: C:\Documents	and Settings\Administrator\Desktop\robodkit	

 4. ให้ทำการตั้งชื่อที่ Project name (ในตอนนี้จะใช้ชื่อว่า robot_1) และให้เลือก Folder ที่เก็บ code ซึ่งจะต้องไปเก็บไว้ใน folder ที่ได้สร้างขึ้นไว้แล้วจากนั้นให้เลือก Next

Project type:	Project name:
Atmel AVR Assembler	robot_1
AVR GCC	Create initial file
	Initial file:
	robot_1 .c
L analian	
Location:	
C:\Documents and Settings\Adminis	trator\Desktop\robodkit

ทำการตั้งชื่อ และเลือกที่เก็บงาน

5. .ให้เลือก AVR Simulator และที่ขวามือจะมีเบอร์ IC ของ AVR มากมาย

	 Select debug platform and device Debug platform: 	Device:	
Shudio 4	AVR Dragon AVR Simulator AVR Simulator V2 (preview) ICE200 ICE40 ICE50 JTAG ICE JTAGICE mkII	ATmega3290 ATmega3290P ATmega329P ATmega32HVB ATmega406 ATmega48 ATmega48P ATmega64 ATmega640 ATmega640 ATmega644 ATmega644P	
	🗖 Open p	olatform options	
/er 4.13.571			

เถือก AVR Simulator

6. ให้หาเบอร์ ATmega8 ซึ่งเป็นเบอร์ที่ใช้กับหุ่นยนต์ดังกล่าว จากนั้นให้เลือก Finish

	Debug platform:	Device:	
Sindio 4	AVR Dragon AVR Simulator AVR Simulator V2 (preview) ICE200 ICE40 ICE50 JTAG ICE JTAGICE mkll	ATmega644 ATmega644P ATmega645 ATmega6450 ATmega6490 ATmega6490 ATmega8 ATmega8515 ATmega8535 ATmega88 ATmega88 ATmega88	
	F Open	platform options	

เลือกเบอร์ ATmega8

7. จะได้หน้าต่างโปรแกรมพร้อมที่จะทำการเขียนโปรแกรม

AVR Studio - C: Documents a	and Settings\Administrator\Desktop\ro	bodkit\robot_1.c					
Ble Broject Build Edit View	Iools Debug Window Help	R R I G I M I M I	a o su la	1 'n 10 🗊 🛛	o vije		J.
Trace Disabled	「十十十二日日日本がある」	BB\$X6	-				
S ⁴ C robot 1 (def witt)	B C: Documents and Settings Udmi	inistrator Wesktop \rob 🖳		1/0 vew		10.0	* *
3 Garce Files 업 Header Files 영 Enternal Dependencies 영 Other Files			I	DAD_CONV DANALOG_ OFU DANALOG_ OFU OF	ERTER COMPARA_ L_INTERR_ DUNTER_0 DUNTER_1 DUNTER_2 DD		-
	1.11		<u>مر</u>	Name	Address	Value Bits	F
Build	C:\Documents and Settings\Adminis	trator\Desktop\robodkit\robo	€1¢ () • X				
Build OMessage Red in File	es Breakpoints and Tracepoints						
		ATmegað	Avil: Smith	star Auto	10	Ln 1, Col 1	CAP NUM OVA

หน้าต่างโปรแรกม

8.ต่อไปจะเป็นขั้นตอนการกำหนดรายละเอียดการคอมไพล์ไฟล์โปรเจ็กต์



ให้ไปที่เมนู Project เลือกรายการ Configuration Option จะได้หน้าต่างดังรูปด้านล่าง

<u>`</u>	Active Configuration default	nfigurations
neral	Use External Makefile 1. Target name must equal project name. 2. Clean/rebuild support requires "clean" target. 3. Makefile and target must exist in the same folder	
ude tories	Output File Name: robot_1.elf Output File Directory: default\	
ies P pry gs	Device: atmega8 ✓ Unsigned Chars (-funsigned Chars) Frequency: 11059200 hz ✓ Unsigned Bitfields (-funsigned Chars) Optimization: -01 ✓ Pack Structure Members (-00 -01 ✓ Short Enums (-fshort-enum)	d-char) ned-bitfields) -fpack-struc s)
•	-02 -03 IV Create Hex F-Os prate Map File IV Gene	rate Li <mark>st Fil</mark> e

Loaded plugin AVR GCC

ในหน้าต่างนี้ในช่อง Frequency ให้กำหนดค่าความถี่เท่ากับ 11059200 hz เพื่อให้ตรงกับความถี่ของตัวหุ่น และในช่อง Optimization คือช่องการเลือกการบีบอัดข้อมูลจะมีอยู่ 4 ระดับ ให้เลือกไว้ที่ –O1 หรือ –Os ก่อนสำหรับ โปรแกรมตัวอย่าง ในกรณีที่ทำงานในโหมดดีบักแล้วไม่เกิดผลใดๆในการทำงานของโปนแกรม ให้เลือกไปที่ –O0 เพื่อ ยกเลิกการบีบอัดข้อมูล

<u>หมายเหตุ</u> ในการเขียนโปรแกรมใหม่ทุกครั้งต้องเข้ามากำหนด Configuration Option เสมอ

9. ถ้าตอนนี้ทำการเปิดดู Folder / robodkit จะพบไฟล์อยู่ 2 ไฟล์ดังภาพดังกล่าว



10. ทำการเขียน Code ในหน้าต่าง Edit ดังนี้



11. จากนั้นให้ ทำการคอมไพล์ เพื่อตรวจสอบการทำงานของ code โดยไปที่เมนู Build > Compile หรือ ใช้ Alt + F7

Project	<u>B</u> uild	d <u>E</u> dit	<u>V</u> iew	<u>T</u> ools	<u>D</u> ebu	ig <u>W</u> indov
3 🛛 🖉		<u>B</u> uild			F7	🙀 i 🖊
Disabled		Rebuild	<u>A</u> ll			Con AVR
-		Build and	d <u>R</u> un	Ctr	I <mark>+F7</mark>	
- obot_1 (d	٢	Compile		Alt	:+F7	uments a
🗟 Source	×	Clean			F12	
🔄 Heade		Export M	Makefile		1	ude <av ain(voi</av
Other I	Gr <i>b</i> o Files	ронаонск		[{ Γ	IDRB	=0%FF·
		o		ין ד <u>י</u> איל		<u></u> ,
		ทา	การคอม	เพล		

12. ถ้าคอมไพล์ผ่าน จะขึ้นลักษณะดังกล่าว คือ เขียน code ถูกต้อง หากเขียนผิดก็ให้ทำการแก้ไขและคอมไพล์ใหม่จนกว่าจะ

External Dependencies Other Files	<pre>int main(void) { DDRB =0XFF; DDRD =0XFF; while(1) { FORTB=0x10; // 0001 0000 PORTD=0x80; // 1000 0000 } }</pre>
	C:\Documents and Settings\Administrator\Desktop\robodkit\robot 1.c 4
Build	• ×
Build succeeded with 0 1	▼ X ga8 -Wall -cdwarf-2 -Os -std=gnu99 -funsigned-char -funsigned-bitfiel Warnings



13. หากเข้าไปดูที่ Folder / robodkit จะเห็นว่าได้ไฟล์ใหม่และ Folder / default เกิดขึ้น

۱	Administrator\Deskto	op\robodkit	
	Name 🔺	Size	Туре
	🚞 default		File Fold
	💌 robot_1	3 KB	aps File
	🗐 robot_1	1 KB	C compil

เข้าไปดูใน Folder / Robodkit

14. จากนั้นให้ทำการ Build โดยไปที่เมนู Build > Build หรือ กด F7

itudio	- C:	Doc	um	ents a	nd S	etting	;s \A d
roject	Build	i E	dit	<u>V</u> iew	Tools	s <u>D</u> eł	bug
		<u>B</u> uild	1			F7	
sabled		Reb	uild	All			3.0
		Build	and	d <u>R</u> un	C	trl+F7	
ot_1 (d	۲	Com	pile		1	Alt+F7	um
Source	×	Clea	n			F12	
Heade		Expo	ortN	1akefile			aiı
External openacional { Other Files			{ החסד	יח- מ			



15. ถ้ำ Build สมบูรณ์จะขึ้นข้อความคังกล่าว

		ia Sectings (A
Build		
(.text + .	data + .pootioader)	
Data:	0 bytes (0.0% Full)	
(.data + .	bss + .noinit)	
Build succ	eeded with O Warnings	
<		

16. ถ้าเข้าไหดูใน Folder / robodkit ใน Folder Default จะพบมีไฟล์ robot_1 .HEX ซึ่งเป็นไฟล์ที่ใช้ในการ load code เข้าไป ในตัว IC ครับ

Name 🔺			
💼 default 🔪			
robot_1			
🗐 robot_1			
aep		Flie Folder	
🔟 Makefile	2 KB	File	
🚾 robot_1	1 KB	Intel HEX file	
📽 robot_1	11 KB	C compiler map	file
🖬 robot_1.elf	5 KB	ELF File	
🔟 robot_1.lss	5 KB	LSS File	
國 robot_1.o	3 KB	O File	

เมื่อ Build แล้วจะใด้ .HEX

เมื่อเสร็จสิ้นขั้นตอนการเขียนโปแกรมแล้วขั้นตอนต่อไปจะทำการ load Program เข้ากับตัวหุ่นยนต์และทคสอบหุ่นยนต์กัน

ขั้นตอนการ Load Program และ การทดสอบหุ่นยนต์

การเตรียมหุ่นยนต์

หุ่นยนต์จะต้องมีการประกอบให้สมบูรณ์และต่อขั้วต่างๆให้ถูกต้อง และต้องใช้ถ่านที่ใหม่ จากนั้นทำการต่อสาย USB เข้ากับคอมพิวเตอร์และตัวหุ่นยนต์ดังภาพ และในระหว่างการ load program จะต้องทำการปิดสวิตช์หุ่นยนต์ก่อนทุก ครั้ง ให้เหลือแค่ไพเลี้ยงจาก USB เท่านั้น



ปิดสวิตช์ก่อนทำการ Load program

การเรียกหา USB Drivers ของหุ่นยนต์เมื่อติดต่อกับกอมพิวเตอร์ในกรั้งแรก

เมื่อต่อสาย USB เข้ากับหุ่นยนต์และเครื่องคอมพิวเตอร์ครั้งแรก เครื่องคอมพิวเตอร์อาจจะยังไม่รู้จักจึงจะมีการเรียก หา USB Drivers ซึ่ง USB Drivers คังกล่าวได้อยู่ในแผ่น CD ที่แถมมาให้แล้ว ใน Folder / FTDI USB Drivers

ขั้นตอนการ Load program

เนื่องจาก เราใช้ PORT USB ในการ load program เราจำเป็นต้องรู้ว่า Port ที่ใช้งานคังกล่าวเป็น Port COM ที่เท่าไร และยังเป็นการทคสอบด้วยว่าการเชื่อมต่อสมบูรณ์หรือไม่ (ในตัวอย่างจะได้ COM 3)



การตรวจสอบการติดต่อกับ USB สมบูรณ์

กลับไปที่ AVR Studio 4 อีกครั้ง และทำตามขั้นตอนดังนี้

1. ไปที่เมนู Tool > Program AVR > Connect...



การเลือกการ Connect

 2. จากนั้นเลือก STK500 or AVRISP ส่วน Port ก็เลือก Port ที่ติดต่อกับ USB หรือ ให้เลือก Auto ก็ได้เพราะจะหาให้เอง จากนั้นเลือก Connect...

Platform:	Port:	
STK500 or AVRISP JTAG ICE JTAGICE mkll	Auto COM1 COM2	Connect
AVRISP mkll AVR Dragon STK600	COM3 COM4 COM5	Baud rate:
ip: To auto-connect to the outton on the toolbar.	programmer used last time, press the 'P	rogrammer' Baud rate changes ar
Note that a tool cannot be u a debugging session. In that	sed for programming as long as it is cor case, select 'Stop Debugging' first.	nnected in

การเลือกการ Connect เพื่อติดต่อกับตัวหุ่นยนต์

3. เมื่อ Connect ผ่านจะเข้าสู่หน้า AVRISP ดังกล่าว

Erase Device	I Erase de I Verify de	evice before programming evice after programming
lash		
Use Current Simulator/Emi Input HEX File	ulator FLASH Memory	
Program	Verify	Read
Use Current Simulator/Em Input HEX File	ulator EEPROM Memory	
Program	Verify	Read
LF Production File Format		
Program	Save	Fuses and lockbits settings must be specified before saving to ELF
sting on 'COM3' SP with V2 firmware found on CC	рмз	

4. ให้เลือกเมนู Main เพื่อกำหนดคุณสมบัติต่างๆ ที่ Device ให้เลือกเบอร์ IC คือ ATmega8

VRISP with top module '0x00' in ISP mode	e with ATmega8 📃 🗖 🔯
Main Program Fuses LockBits Advanced H	HW Settings HW Info Auto
Device and Signature Bytes	
ATmega8	Erase Device
Signature not read	Read Signature
Programming Mode and Target Settings	Settings
	ISP Frequency: 921.6 kHz

หน้าต่าง Main

จากนั้นให้กด Read Signature เพื่อที่จะทำการติดต่อกับ IC ส่วนก่าอื่นๆให้ไว้คงเดิม

Tmega8	Erzee Deulee
x1E 0x93 0x07	Bead Signature
mature matches selecte	d device
duarate marches selecte	a avilar
gnature matches selecte	
ogramming Mode and Ta	rget Settings
ogramming Mode and Ta P mode	rget Settings

หน้าต่าง Main แสดงการอ่าน Read Signature

5. เมื่อเข้ามาดูที่ Fuses จะได้การกำหนดเงื่อนไขดังรูป ไม่ต้องไปเปลี่ยนแปลง

WTDON SPIEN		
SPIEN	ETC.	
EESAVE		
BOOTSZ	Boot Flash size=1024 words Boot address=\$0C00	
BOOTRST	✓	
CKOPT		
BODLEVEL	Brown-out detection at VCC=2.7 V	3
BODEN		
SUT_CKSEL	Ext. Crystal/Resonator High Freq.; Start-up time: 1K CK + 0 ms	1
HIGH	0xC8	
LOW	0xEE	

หน้าต่าง Fuses

6. จากนั้นให้เข้ามาที่หน้า Program ในส่วนของ Flash ให้เลือก Input HEX File และทำการเลือก ไฟล์ที่จะทำการ load (ในที่นี้ คือ robot_1.HEX)

	Lockbills Advanced Hw Settings Hw Inio Adio
Device	L Frace device before programming
Erase Device	✓ Urase device before programming
Flash	
C Use Current Sim	nulator/Emulator FLASH Memory
Input HEX File	tings\Administrator\Desktop\robodkit\default\robot_1.hex
Program	Verify Read
FEPBOM	
C Use Current Sim	nulator/Emulator EEPROM Memory
Input HEX File	
Program	Verify Read
	mat
ELF Production File Fo	and.
ELF Production File For Input ELF File	
ELF Production File For Input ELF File	Fuses and lockbits settings

7. เมื่อเลือกไฟล์แล้วให้ทำการเลือกที่ Program จากนั้น code ต่างๆก็จะเข้าสู่ตัว IC ในหุ่นยนต์ โดยส่วนล่างจะเป็นการ แสดงผลการทำงาน

n riogiani Fuses Loc	KBits Advanced Hw Setti	ngs Hwinno Auto
Erase Device	I⊄ Erase d I⊄ Verify di	evice before programming evice after programming
Flash		
Use Current Simulator Input HEX File Iting:	/Emulator FLASH Memory s\Administrator\Desktop\robo	dkit\default\robot_1.hex
Program	Verify	Read
Ose Current Simulator Input HEX File	Emulator EEPHUM Memory	
Program	Verify	Read
ELF Production File Format		
Input ELF File		
Program	Save	Fuses and lockbits settings must be specified before saving to ELF
1.1.084		

การ Program และ รายละเอียดการทำงาน

8. จากนั้นให้ทำการทอดสาย USB ออก และทำการเปิดสวิตช์หุ่นยนต์ หุ่นยนต์จะวิ่งไปข้างหน้า แสดงว่า code ที่เขียนถูกต้อง และการ load program สมบูรณ์



พร้อมวิ่งแล้วครับ...

ส่วนน้องๆที่สนใจเกี่ยวกับการเขียนโปแกรมของหุ่นยนต์ตัวนี้สามารถอ่านได้ในบทความนะครับ ซึ่งจะมี Code อื่นๆ อีกมากกว่า เช่น หุ่นยนต์เดินตามเส้น หุ่นยนต์หลบหลีกสิ่งกีดขวาง หุ่นยนต์ซูโม่ และอื่นๆ ลองติดตามในบทความนะครับ พี่ๆทีมงานจะเขียนไว้ให้อ่านกันครับ

ตางรางรูปแบบการเคลื่อนที่ของหุ่นยนต์				
รูปแบบการเคลื่อนที่	Code / คำสั่งในการเขียนโปรแกรม			
	PORTB = 0x00			
หุ่นขนตหยุดอยู่กบท	PORTD = 0x00			
***********	PORTB = 0x10			
ពុំដល់ដាមាជាជា	PORTD = 0x80			
น่าเยาเต็ออยนอัง	PORTB = 0x20			
นี้หอหมเกอบเป	PORTD = 0x40			
3/31/213 (2020) 3/31	PORTB = 0x10			
น้ากดหมุ่มที่ทุกบ	PORTD = 0x40			
323161312346913129761	PORTB = 0x20			
น้ากดหมุ่มที่หญาด	PORTD = 0x80			
าร่างและเต้เลื่อเวลเวา	PORTB = 0x10			
นี้ หากหมเยก 1 ม 1 เ	PORTD = 0x00			
ala 100 1 2 3 0 2 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	PORTB = 0x00			
นี้หอหมเนอ เฉ เอ	PORTD = 0x80			
**************************************	PORTB = 0x00			
มี่ หอหผุยคอมยุ่งเยอ 1.0 1.1	PORTD = 0x40			
	PORTB = 0x20			
มี่ หอหผุยคอมยุ่งเยอ เจ เอ	PORTD = 0x00			
รูปแบบการรับสัญญาณของ sensor				
Sensor L อยู่ที่พื้น สีดำ	(PIND&(1< <pind2)) 0<="" =="" td=""></pind2))>			
Sensor L อยู่ที่พื้น สีขาว	(PIND&(1< <pind2)) 1<="" =="" td=""></pind2))>			
Sensor R อยู่ที่พื้น สีดำ	(PINC&(1< <pinc2)) 0<="" =="" td=""></pinc2))>			
Sensor R อยู่ที่พื้น สีขาว	(PINC&(1< <pinc2)) 1<="" =="" td=""></pinc2))>			

ตัวอย่างโปรแกรมการทดลอง

1 :		/*	
2: 3 .		Workfile : Robot1.c	
4 :		,	
5 :		<pre>#include <avr io.h=""></avr></pre>	//เป็นการเรียกใช้งานรีจิสเตอร์และฟังชันต่าง
6 :			
7 :		int main(void)	//ฟังชันหลักของภาษาซี
8 :		{	
9 :		DDRB =0XFF;	// กำหนดให้ Port B เป็น output ทั้งหมด
10	:	DDRD =0XFF;	// กำหนดให้ Port D เป็น output ทั้งหมด
11	:		
12	:	while(1)	// เป็นคำสังวนลูปในภาษาซี จะวนการทำงานในบรรทัดที่ 12-16
13	:	{	
14	:	PORTB=0x10;	//กำหนดให้มอเตอร์ขวาเดินหน้า
15	:	PORTD=0x80;	//กำหนดให้มอเตอร์ซ้ายเดินหน้า
16	:	}	
17	:	}	

จากโปรแกรมข้างค้นเราสามารถกำหนดค่าของ PortB และ PortD ตามตารางที่ให้ไว้ค้านบน และลองดูการเปลี่ยนแปลงการ ทำงานของตัวหุ่นว่าเป็นไปตามที่ตารางกำหนดไว้หรือไม่ ตัวอย่างต่อไปเราจะเรียกใช้ฟังชันหน่วงเวลาหรือ #include <util/delay.h> มาดูตัวอย่างกันเลยครับ

```
/*
  :
1
       Workfile
                      : Robot2.c
2
  :
       */
3
  :
                                            //เป็นการเรียกใช้งานรีจิสเตอร์และฟังชันต่าง
       #include <avr/io.h>
4
  :
                                            //เรียกใช้งานฟังชันหน่วงเวลา
       #include <util/delay.h>
5
  :
                                            //กำค่าความถี่ที่ใช้ในการหน่วงเวลา 11059200 hz
       #define F_CPU 11059200UL
6
  :
7
  :
       /********** delay **************/
8
  :
       void delay_ms(unsigned int i)
                                            //ฟังชั้น หน่วงเวลา
9 :
10 :
       {
11 :
              for(; i>0; i--)
12
    :
                      _delay_ms(1);
13 :
       }
14
    :
       15
                                            //ฟังชั่นหลักของภาษาซึ่
16
    :
       int main(void)
17
    :
       {
                                            // กำหนดให้ Port B เป็น output ทั้งหมด
18
   :
              DDRB =0XFF;
                                            // กำหนดให้ Port D เป็น output ทั้งหมด
19 :
              DDRD =0XFF;
20
    •
                                            // เป็นคำสังวนลูปในภาษาซี จะวนการทำงานในบรรทัคที่ 12-16
21
              while(1)
   :
    :
22
                                            //กำหนดให้มอเตอร์ขวาเดินหน้า
                      PORTB=0x10;
23 :
                                            //กำหนดให้มอเตอร์ซ้ายเดินหน้า
24
    :
                      PORTD=0x80;
                                            //หน่วงเวลาการทำงาน 500มิลลิวินาที หรือ 0.5 วินาที
25
    :
                      _delay_ms(500);
                      PORTB=0x00;
                                            //กำหนดให้มอเตอร์ขวาหยุด
26
    :
                                            //กำหนดให้มอเตอร์ซ้ายหยุด
27
   :
                      PORTD=0x00;
                                            //หน่วงเวลาการทำงาน 500มิลลิวินาที หรือ 0.5 วินาที
28
   :
                      _delay_ms(500);
29 :
                      }
30 :
       }
```

จากโปรแกรม Robot2 จะเห็นว่ามอเตอร์จะทำงานแบบเดินหน้าไป 0.5 และหยุด 0.5 วินาทีสลับกันไป

โจทย์

1.เพิ่มเวลา จาก 500 เป็น 1000 ดูการเปลี่ยนแปลง

2.ให้มอเตอร์เดินหน้า 500 และถอยหลัง 500

3.ให้มอเตอร์เดินหน้า 500 เลี้ยวขวา 500 เลียวซ้าย 500 ถอยหลัง 500 หยุด 500

้ตัวอย่างต่อไปจะเป็นการใช้งานตัวตรวจจับสัญญาณ input หรือ sensor /* 1 2 Workfile : Robot2.c : */ 3 : //เป็นการเรียกใช้งานรีจิสเตอร์และฟังชันต่าง #include <avr/io.h> : 5 : 6 : 7 : int main(void) //ฟังชั่นหลักของภาษาซึ 8 : { // กำหนดให้ Port B เป็น output ทั้งหมด 9 : DDRB =0XFF; // กำหนดให้ Port D7,D6 เป็น output D5-D0 เป็น input 10 : DDRD =0XC0; // กำหนดให้ Port D เป็น input ทั้งหมด DDRC =0X00; 11 12 • // เป็นคำสังวนลูปในภาษาซี จะวนการทำงานในบรรทัดที่ 12-16 while(1) 13 : 14 : { //ตรวจจับ sensor ซ้าย มีการทำงาน If((PIND&(1<<PIND2)) != 0) 15 : 16 //กำหนดให้มอเตอร์ซ้ายเดินหน้า PORTD=0x80; 17 : : } 18 //ตรวจจับ sensor ซ้าย มีการทำงาน else 19 : 20 : { //กำหนดให้มอเตอร์ซ้ายหยุด PORTD=0x00; 31 : 32 : } 33 If((PINC&(1<<PINC2)) != 0) //ตรวจจับ sensor ขวา มีการทำงาน 21 : 22 : //กำหนดให้มอเตอร์ขวาเดินหน้า PORTB=0x10; 23 : 24 : } //ตรวจจับ sensor ขวา มีการทำงาน 25 : else 26 : //กำหนดให้มอเตอร์ขวาหยุด PORTB=0x00; 34 • 35 : } 36 : } 37 : }

เมื่อโหลดโปรแกรมลงตัวหุ่นแล้ว ให้ลองเอามือเข้าไปใกล้กับตัว sensor ทั้ง ซ้ายและขวา จะสังเกตเห็นว่า มอเตอร์จะหมุน ในขณะที่ sensor มาอยู่ใกล้มือเรา เท่านี้เราก็ได้หุ่นยนต์เดินตามเส้นแล้วครับ

ตัวอย่าง SUMO ROBOTKIT

/**************************************	*****	
* Workfile : AVR	_SUMO_1 Test motor	*
* Purpose : DC M	Aotor Control	*
* Copyright : Advance Project Group Co.,Ltd.		*
* Author : Santichai jaichuen		*
* Email : santi@robodkit.com		*
* Compiler : AVR Studio/WINAVR		*
* Target : Atmega8A	;	*
* Other Files : PCB AP107-1	,	*
******	****************	
#include <avr io.h=""></avr>		
#include <compat deprecated.h=""></compat>	// Use sbi(), cbi() function	
<pre>#include <util delay.h=""></util></pre>		
#define F_CPU 11059200UL	//11.0592 MHz	
/**************************************	**************	
#define DC_MR_POUT	PORTB	
#define DC_MR_DDR	DDRB	
#define DC_MR_DDR_OUT	(1< <ddb4) (1<<ddb5)< td=""><td></td></ddb4) (1<<ddb5)<>	
#define DC_ML_POUT	PORTD	
#define DC_ML_DDR	DDRD	
#define DC_ML_DDR_OUT	(1< <ddd6) (1<<ddd7)< td=""><td></td></ddd6) (1<<ddd7)<>	
#define MOTOR_RIGHT_F	4 // PIN PB4	
#define MOTOR_RIGHT_B	5 // PIN PB5	
#define MOTOR_LEFT_B 6	// PIN PD6	
#define MOTOR_LEFT_F 7	// PIN PD7	
<pre>#define ON_MR(x) sbi(DC_MR_POUT, x);</pre>		
<pre>#define OFF_MR(x) cbi(DC_MR_POUT, x);</pre>		
<pre>#define ON_ML(x) sbi(DC_ML_POUT, x);</pre>		
<pre>#define OFF_ML(x) cbi(DC_ML_POUT, x);</pre>		
/****************** delay_ms *	***************************************	
void delay_ms(uint16_t i)		
{		
for (; i>0; i)		
_delay_ms(0.65);		

}

```
void Motor_Forward()
{
     ON_ML(MOTOR_LEFT_F);
                                    // Motor left start
     OFF_ML(MOTOR_LEFT_B);
     ON_MR(MOTOR_RIGHT_F);
                                    // Motor right start
     OFF_MR(MOTOR_RIGHT_B);
}
void Motor_Backword()
{
     OFF_ML(MOTOR_LEFT_F);
                                    // Motor left start
     ON_ML(MOTOR_LEFT_B);
     OFF_MR(MOTOR_RIGHT_F);
                                    // Motor right start
     ON_MR(MOTOR_RIGHT_B);
}
void Motor_Stop()
{
     OFF_ML(MOTOR_LEFT_F);
                                    // Motor left start
     OFF_ML(MOTOR_LEFT_B);
     OFF_MR(MOTOR_RIGHT_F);
                                    // Motor right start
     OFF_MR(MOTOR_RIGHT_B);
}
void Motor_Turn_Left()
{
     ON_ML(MOTOR_LEFT_F);
                                    // Motor left start
     OFF_ML(MOTOR_LEFT_B);
     OFF MR(MOTOR RIGHT F);
                                    // Motor right start
     OFF_MR(MOTOR_RIGHT_B);
}
void Motor_Turn_Right()
{
     OFF_ML(MOTOR_LEFT_F);
                                    // Motor left start
     OFF_ML(MOTOR_LEFT_B);
```

```
ON_MR(MOTOR_RIGHT_F);
                                           // Motor right start
      OFF_MR(MOTOR_RIGHT_B);
}
void Motor_Rotate_Left()
{
      ON_ML(MOTOR_LEFT_F);
                                           // Motor left start
      OFF_ML(MOTOR_LEFT_B);
      OFF_MR(MOTOR_RIGHT_F);
                                           // Motor right start
      ON_MR(MOTOR_RIGHT_B);
}
void Motor_Rotate_Right()
{
      OFF_ML(MOTOR_LEFT_F);
                                           // Motor left start
      ON_ML(MOTOR_LEFT_B);
      ON_MR(MOTOR_RIGHT_F);
                                           // Motor right start
      OFF_MR(MOTOR_RIGHT_B);
}
/************* Main Functions ****************/
int main(void)
{
      DC_MR_DDR = DC_MR_DDR_OUT;
                                       // Set Port Output
      DC_ML_DDR = DC_ML_DDR_OUT;
                                                // Set Port Output
 while (1){
      Motor_Forward();
      delay_ms(1000);
      Motor_Stop();
      delay_ms(50);
      Motor_Backword();
      delay_ms(1000);
      Motor_Stop();
      delay_ms(50);
```

Motor_Turn_Left(); delay_ms(2000); Motor_Stop(); delay_ms(50);

Motor_Turn_Right();

delay_ms(2000);

Motor_Stop();

delay_ms(50);

Motor_Rotate_Left();

delay_ms(2000);

Motor_Stop();

delay_ms(50);

Motor_Rotate_Right(); delay_ms(2000); Motor_Stop(); delay_ms(50);

}

return 0;