Lab Lecture 2 – Fall 2018 Arduino

Intelligent Robotics Lab

Ph.D. Husnu Melih Erdogan – herdogan@pdx.edu Lab Assistant





Introduction to Arduino



Who has any experience with Arduino?



Lecture Overview

• What is Arduino?



What is Arduino?

- Arduino is an open-source platform used for building electronics projects.
- Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. (learn.sparkfun.com)







Where can I buy one?

- You can get your Arduino board online on their website, or any other shopping websites such as Amazon, eBay, Adafruit, and Element14.
- It is an open-source project, so there are many Arduino like boards on the Internet. They all should work the same.





Arduino Specs

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g



Arduino Family (Most Popular)



Uno



Mega



Lily Pad

MKR1000 WIFI



Micro



Nano



MKR Vidor 4000



What is on the board?



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.



IO pins on Arduino

- The I/O pins provide 5V 20mA
- The 3.3V supply is limited to 50mA.
- Using the IO outputs is an easy task. They are on or off. In other words, the signal on a IO output pin can be HIGH or LOW.
- HIGH means 5V and LOW means 0v.



Arduino Pinouts













A Simple circuit design that uses the Arduino GPIO Pins.



• The voltage of an LED is between 1.8V and 3.3V.

$$R = \frac{\Delta V}{I} = \frac{3V - 2.2V}{20mA} = 40\Omega$$
 5 - 2.2 / 0.02 = 140 Ω





Arduino FPGA



https://youtu.be/v9lln2mYdlo?t=4

https://www.arduino.cc/en/Tutorial/VidorHDL



Arduino Shields

- Attachable hardware to the Arduino GPIO pins.
- They are used to extend Arduino's functionality
- There are so many different Arduino shields.





What you will need

- USB Cable
- 5V Power Supply
- Battery Holder
- Arduino IDE or a web browser
- Computer





Blink | Arduino 1.8.7 (Windows Store 1.8.15.0)

Taxas as IFD on for one second, then off for one second, repeated)

modified 8 May 2014 by Scott Fitzgerald modified 2 Sep 2016 by Arturo Guadalupi modified 8 Sep 2016 by Colby Newman

is example code is in the public domain

p://www.arduino.cc/en/Tutoria

// the setup function runs once when you press reset or power the boar void setup() (// initialise digital pin LED_BUTLTIN as an output. pinktode(LED_BUTLTIN, OUTFOT);

// the loop function runs over and over again forever exist loop() (digitalWriteLEA_WITTH, HIGH); // turn the LED on (HIGH is the voltage level) digitalWriteLEA_WITTH, LOW]; // turn the LED off by making the voltage LOW digitalWriteLEA_WITTH, LOW]; // turn the LED off by making the voltage LOW digitalWriteLEA_WITTH, and the second







Arduino IDE - https://www.arduino.cc/en/Main/Software



Arduino Web Editor

You can use Arduino on your browser if you install their plugin for your Internet Browser.

	×	Blink	0
> EDITOR	SEARCH EXAMPLES Q	✓ → Select Board or Port ··· SHARE	
C Sketchbook	SHOWING EXAMPLES FOR ALL BOARDS	Blink.ino Blink.txt	
Examples	BUILT IN FROM LIBRARIES	2 Blink 3 4 Turns an LED on for one second, then off for one second, repeatedly.	K 2
💾 Libraries	01.BASICS (6)	5 6 Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO 7 it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to 8 the compact LED pin independent of which board is used	
Q. Monitor	AnalogReadSerial	9 If you want to know what pin the on-board LED is connected to on your Arduino model, check the Technical Specs of your board at:	
⑦ Help	BareMinimum	11 https://www.arduino.cc/en/Main/Products 12 modified & Max 2014	
	🖺 Blink 🔘	14 by Scott Fitzgenald 15 modified 2 Sep 2016	
	DigitalReadSerial	17 modified 8 Sep 2016 18 by Colby Newman	
	🖺 Fade	19 20 This example code is in the public domain.	
	ReadAnalogVoltage	<pre>22 http://www.arduino.cc/en/Tutorial/Blink 23 */ 24</pre>	
	02.DIGITAL (9)	<pre>25 // the setup function runs once when you press reset or power the board 26 • void setup() {</pre>	
	03.ANALOG (6)	<pre>27 // initialize digital pin LEU_BULLIN as an output. 28 pinMode(LED_BUILTIN, OUTPUT); 29 } 29 }</pre>	
	04.COMMUNICATION (12)	30 31 // the loop function runs over and over again forever 32 • void loop() {	
	05.CONTROL (6)	33 digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level) 34 delay(1000); // wait for a second 35 digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW	
	06.SENSORS (4)	<pre>36 delay(1000); // wait for a second 37 } 38</pre>	
	07.DISPLAY (2)		
	08.STRINGS (14)		ß
web services	09.USB (7)		



Some of the good sources to learn more about Arduino

- Arduino Tutorials
- Adafruit Website
- Hack a Day
- SparkFun



Introduction to Arduino Programming Language



Arduino Programming Language

- The Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Arduino is great for prototyping.
- There are a lot of libraries that allow you to use many different devices with your Arduino.

https://www.arduino.cc/reference/en/

Libraries:

The Arduino environment can be extended through the use of libraries, just like most programming platforms. Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data.

To use a library in a sketch, select it from Sketch > Import Library.

https://www.arduino.cc/en/Reference/Libraries



Arduino Functions

Digital I/O digitalRead() digitalWrite() pinMode()

Analog I/O analogRead() analogReference() analogWrite()

Zero, Due & MKR Family analogReadResolution() analogWriteResolution()

Advanced I/O noTone() pulseIn() pulseInLong() shiftln() shiftOut() tone()

cos() sin() tan() Characters

Trigonometry

isAlpha()

isAscii()

isDigit()

isControl()

isAlphaNumeric()

isGraph() isHexadecimalDigit() isLowerCase()

> isPrintable() isPunct()

isSpace() isUpperCase()

isWhitespace()

Characters isAlpha() isAlphaNumeric() isAscii() isControl() isDigit() isGraph() isHexadecimalDigit() isLowerCase() isPrintable() isPunct() isSpace() isUpperCase()

Time

delay()

micros()

millis()

delayMicroseconds() isWhitespace()

random() randomSeed() Bits and Bytes bit() bitClear() bitRead() bitSet() bitWrite() highByte() lowByte() **External Interrupts** attachInterrupt() detachInterrupt()

Random Numbers

Interrupts interrupts() noInterrupts()



USB Keyboard

Mouse



Variable

Constants
Floating Point Constants
Integer Constants
HIGH I LOW
INPUT I OUTPUT I INPUT_PULLUP
LED_BUILTIN
true l false
Conversion
byte()
char()
float()
int()
long()
word()

String() array bool boolean byte char double float int long short string unsigned char unsigned int unsigned long void word

Variable Scope & Qualifiers

const

scope

static volatile

Utilities PROGMEM sizeof()



Structural Elements

Sketch
loop()
setup()
Control Structure
break
continue
dowhile
else
for
goto
ifelse
return
switchcase
while

Further Syntax #define (define) #include (include) /* */ (block comment) // (single line comment) ; (semicolon) {} (curly braces)

Portland State

Operators

Arithmetic Operators

- % (remainder)
- * (multiplication)
- + (addition)
- (subtraction)
- / (division)
- = (assignment operator)

Comparison Operators

- != (not equal to)
- < (less than)
- <= (less than or equal to)
- == (equal to)
- > (greater than)
- >= (greater than or equal to)

Boolean Operators

- ! (logical not)
- && (logical and)
- ll (logical or)

- Pointer Access Operators
- & (reference operator)
- * (dereference operator)
- **Bitwise Operators**
- & (bitwise and)
- << (bitshift left)
- >> (bitshift right)
- ^ (bitwise xor)
- l (bitwise or)
- ~ (bitwise not)

Compound Operators

- **&=** (compound bitwise and)
- *= (compound multiplication)
- ++ (increment)
- += (compound addition)
- -- (decrement)
- -= (compound subtraction)
- /= (compound division)
- ^= (compound bitwise xor)
- l= (compound bitwise or)



Conditional Statements – If, elif, else

- The **if** statement contains a logical expression using which data is compared and a decision is made based on the result of the comparison.
- An **else** statement can be combined with an **if** statement. An **else** statement contains the block of code that executes if the conditional expression in the if statement resolves to 0 or a FALSE value.
- The **elif** statement allows you to check multiple expressions for TRUE and execute a block of code as soon as one of the conditions evaluates to TRUE.



if, else if, else



Loops – While, For

For Loop

While Loop





Import Libraries

- There are three different methods to install Arduino • libraries.
 - Library Manager 1.
 - Library Manager with .zip file 2.
 - Manuel Installation 3.

💿 Blink Ar	rduino 1.8.7 (Windows Store	1.8.15.0)				
	Verify/Compile Upload	Ctrl+R Ctrl+U r Ctrl+Shift+U				
// the	Export compiled Binary	Ctrl+Alt+S	you press	reset or p	ower the	board
void	Show Sketch Folder	Ctrl+K	Manage Lib	raries	Ctrl+Shift+I	
//	Add File		Add 7IP Lib	varies	carronner	
}			Arduino libr	aries		
<pre>// the l void loc digita delay digita delay }</pre>	Loop function runs op() [llWrite(LED_BUILTIN (1000); alWrite(LED_BUILTIN (1000);	over and c I, HIGH); I, LOW);	Bridge EEPROM Esplora Ethernet Firmata GSM HID Keyboard LiquidCrysta Mouse Robot Contr Robot IR Re Robot Moto SD SPI Servo SoftwareSer SpacebrewY Stepper TFT Temboo WiFi Wire	al rol mote or ial 'un		he voltage level) r the voltage LOW
			Contributed ArduinoJsor ESP Async V ESP AsyncTO FauxmoESP PubSubClier	l libraries n WebServer CP nt		
			Recommend Adafruit Circ Adafruit MP Adafruit Net	ded libraries cuit Playground vR121 oPixel		





Audi Allov	o Built-In by Arduino vs playing audio files	Version from an	1.0 INSTALLE	D Arduino DU	E on	y. With this library you can use the Ar	duino Due DAC outputs	to play
audi The	o files. audio files must be ir	the raw	.wav format.					
MOLE	Into							
Deci-I	a hu Anduina							
Enat	Je by Arduno les the communications to the characteristics	on betwe	en the Linux	processor a	nd ti	ne AVR. For Arduino Yún and TRE only	. The Bridge library feat	ure:
SD c More	ard, enstablish http c info	ients or s	servers.	mux proces	ses,	open a remote console, access to the	nnux me system, indu	ung the
							Version 1 🔻 I	nstall
EEP	ROM Built-In by Ardui	no, Ch r is	topher Andre	ws Version	2.0	INSTALLED	Version 1.2	
Enat More	Enables reading and writing to the permanent board storage. For all Arduino boards BUT Arduino DUE. <u>More info</u>				Version 1.1			
							Version 1.0	
Espl	ora Built-In by Arduir	o Versio	1.0 INSTAL	LED				
	ts easy access to the	e various ovstick wi	sensors and a th center pus	h-button,4	r the push	Esplora. For Arduino Esplora only. The -buttons,microphone, light sensor, ter	ne sensors available on mperature sensor, 3-axi	the is

UNIVERSITY

- Libraries are often distributed as a ZIP file or folder.
- The name of the folder is the name of the library.
- Inside the folder will be a .cpp file, a .h file and often a keywords.txt file, examples folder, and other files required by the library.
- Starting with version 1.0.5, you can install 3rd party libraries in the IDE.
- Do not unzip the downloaded library, leave it as is.





S Music



Open

Cancel

- When you want to add a library manually, you need to download it as a ZIP file, expand it and put in the proper directory.
- The ZIP file contains all you need, including usage examples if the author has provided them.
- The library manager is designed to install this ZIP file automatically as explained in the former chapter, but there are cases where you may want to perform the installation process manually and put the library in the *libraries* folder of your sketchbook by yourself.
- You can find or change the location of your sketchbook folder at *File > Preferences > Sketchbook* location.


Import Libraries – Method 3

Preferences		×	↓ 🖓 🔜 🖛 Ada	afruit-GF.	X-Library-master		1	>	<
Settings Network			File Home	Share	View			~	0
Sketchbook location: C: \Users\Simone \Documents	Arduino	Browse	← → ~ ↑ 🚺	→ This	PC > Downloads > Adafruit-GFX-Librar	y-master > 🗸 ひ	Search Adafruit-	GFX-Library 🔎	,
Editor language:	English (English) v (requires restart of Arduino)		🖈 Quick access		Name	Date modified	Туре	Size	
Editor font size:	16		Desktop	#	Adafruit-GFX-Library-master	2/6/2017 11:41 PM	File folder		
Interface scale:	Automatic 100 + % (requires restart of Arduino)		Downloads	*					
Show verbose output during: Compilation upload		Desuments							
Compiler warnings:	None 🗸		Documents	71					
Display line numbers		Sten 1	Pictures	A				Sten ?	२
Enable Code Folding			👌 Music					Juch -	,

↓ 🕑 📙 =		Compressed Folder Tools D	ownloads	- 0	×	🔄 📝 📑 🖛 🛛 libraries				- 0	×
File Home Share View Extract File						File Home Share View 🗸 🌔					~ 🕐
← → ~ ↑ ↓ > This PC > Downloads > v ♂ Search Downloads >					$\leftrightarrow \rightarrow \sim \uparrow \square \rightarrow$	This PC > Documents > Arduino > libraries	v 7	Search libraries		م	
📌 Quick access		Name	Date modified	Туре	Size	This PC	Name	Date modified	Type	Size	Î
Desktop	*	📲 Adafruit-GFX-Library-master.zip	2/6/2017 11:25 PM	Compressed (zipped).	(e) (11	
J Downloads	A	MyDownloads	2/6/2017 11:46 PM	File folder		Desktop	readme.txt	2/16/2016 4:50 AM	Text Document	1.6	KB
	*					🔮 Documents	Adafruit-GFX-Library-master	2/6/2017 11:43 PM	File folder		-
Pictures	*			Ste	p 2	Arduino				Stor	ר ר



Serial Communication - UART

- All Arduino boards have at least one serial port (also known as a UART) Serial.
- It communicates on digital pins (RX) and (TX) as well as with the computer via USB.
- If you use these functions, you cannot also use these pins for digital input or output.
- You can use the Arduino environment's built-in serial monitor to communicate with an Arduino board.
- Click the serial monitor button in the toolbar and select the same baud rate used in the call to **Serial begin()**.
- Serial communication on pins TX/RX uses TTL logic levels (5V or 3.3V depending on the board).
- Don't connect these pins directly to other boards that use different logic levels.



Voltage Divider - 5V <-> 3.3V

- There is a problem when connecting a 5V device to 3.3 volt device.
- Therefore, we need to change the logic level from 5V to 3.3V.
- In our case voltage divider is used to connect a digital signal from a 5V device (Arduino) to a GPIO pin on the Pi, which requires a 3.3V input level.
- Raspberry Pi uses 3.3V logic level to decrease the power consumption.





You can build your own voltage divider or you can buy a voltage divider circuit board.



Serial Communication - UART





int potPin =6;

```
void setup () {
    analogReference(DEFAULT);
    Serial.begin(9600);
```

```
void loop(){
   Serial.println(analogRead(potPin));
   delay(500);
}
```



Serial Communication – Serial Monitor



How to load and run Arduino script?

1- Select your board





How to load and run Arduino script?

• 2- Select USB Port

💿 Blink Arduino 1.8.7 (Windows Store 1.8.15.0)								
File Edit Sketch Tools Help								
	Auto Format	Ctrl+T						
	Archive Sketch							
Blink	Fix Encoding & Reload							
	Manage Libraries	Ctrl+Shift+I						
// the set	Serial Monitor	Ctrl+Shift+M	set or power the board					
void setup	Serial Plotter	Ctrl+Shift+L						
// initi	WiFi101 Firmware Undater		put.					
pinMode(Willion Hilliware opdater							
}	Board: "Arduino/Genuino Uno"	>						
	Port: "COM4 (Arduino/Genuino Uno)"	>	Serial ports					
// the loop	Get Board Info		 COM4 (Arduino/Genuino Uno) 					
void loop(Programmer: "AVRISP mkll"	>						
digitalW	Burn Bootloader		LED on (HIGH is the volt	cage level)				
delay(1000)	; /	/ wait for	a second					
<pre>digitalWrite(LED_BUILTIN, LOW); // turn the</pre>		e LED off by making the voltage LOW						
delay(1000); // wait for		r a second						
}								



How to load and run Arduino script?

• 3- Click on upload button







Basic Electronics Overview



Digital Signals

- A digital signal refers to an electrical signal that is converted into a pattern of bits. Unlike an analog signal, which is a continuous signal that contains time-varying quantities, a digital signal has a discrete value at each sampling point.
- Digital signals must have a finite set of possible values. Most commonly digital signals will be one of two values – like either OV or 5V. Timing graphs of these signals look like square waves.





Digital Input

- IO **outputs** are easy; they are on or off, HIGH or LOW, 5v or 0v.
- Inputs are a bit trickier because of the way that digital devices work.
- Although it might seem reasonable just to connect a button across an input pin and a ground pin, the Arduino can get confused as to whether the button is on or off. It might work properly, it might not.
- It's a bit like floating about in deep space; without a reference it would be hard to tell if you were going up or down, or even what up or down meant!
- When a GPIO pin is in input mode and not connected to 5 volts or ground, the pin is said to be **floating**, meaning that it has no fixed voltage level.



Pull Down

Wire the GPIO pin to ground through a large $(10k\Omega)$ resistor so that it always reads LOW. Then you can short the pin to 5 volts by closing the circuit so that it goes HIGH. When the circuit is closed, there is a lower resistance path to 5 volts, and therefore the pin will read HIGH.





Pull Up

Wire a digital IO pin to 5 volts through a large ($10k\Omega$) resistor so that it always reads HIGH. Then we can short the pin to ground by closing the circuit so that the pin will go LOW.





Digital Output

- Using the digital outputs is an easy task.
- They are on or off.
- In other words, the signal on a IO output pin can be HIGH or LOW. HIGH means 3.3v and LOW means 0v.



PWM

- Modulation with the width of an impulse.
- It is used in telecommunication, audio equipment, lights, dc motors, servo motors, etc.
- PMW operates like a switch constantly cycle on and off, so we can control the amount of the power our device get.
- We can control speed of dc motors since, motor takes time to respond to the voltage change, so we take an advantage of duty cycle.



Duty Cycle



Percentage of on time is called duty cycle.

Duty Cycle =Turn ON time/ (Turn ON time + Turn OFF time)

Duty Cycle = (0.5/(0.5+0.5)) = 50%

So the average output voltage will be 50% of the input voltage voltage.



ADC

- ADC stands for Analog to Digital Converter
- Arduino comes with a 10 bit Analog Digital Converter.
- An ADC converts a continuous-time and continuous amplitude (analog signal) to a discrete-time and discrete-amplitude (digital signal).



Breadboard









Lab Activity



Demo 1 – Digital Output Hello World - LED Blink



Digital Output - Blink a LED - Circuit Design





Digital Output - Blink a LED - Code

Blink | Arduino 1.8.7 (Windows Store 1.8.15.0)

File Edit Sketch Tools Help





Demo 2 – Digital Input Push Buttons



Digital Input - Push Button Switch

- A push button is a momentary or non-latching switch which causes a temporary change in the state of an electrical circuit only while the switch is physically actuated.
- An automatic mechanism (i.e. a spring) returns the switch to its default position immediately afterwards, restoring the initial circuit condition.





Digital Input - Push Button Switch – Circuit Design





Digital Input - Push Button Switch – Code

```
💿 button | Arduino 1.8.7 (Windows Store 1.8.15.0)
File Edit Sketch Tools Help
        \checkmark
 button
int switchPin1 = 7;
int switchPin2 = 6;
int ledPin1 = 5;
int ledPin2 = 4;
boolean ledOn1 = false;
boolean ledOn2 = false;
boolean lastButton1 == LOW;
boolean lastButton2 == LOW;
void setup() {
pinMode(switchPin1, INPUT);
pinMode(switchPin2, INPUT);
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
}
void loop() {
if (digitalRead(switchPin1) == HIGH && lastButton1 == LOW ) {
ledOn1 = !ledOn1;
 }
else if (digitalRead(switchPin2) == HIGH && lastButton2 == LOW ) {
ledOn2 = !ledOn2;
}
else{
lastButton1 = digitalRead(switchPin1);
lastButton2 = digitalRead(switchPin2);
}
digitalWrite(ledPin1, ledOn1);
digitalWrite(ledPin2, ledOn2);
delay(50);
```

}





Demo 3 – PWM



PWM – Circuit Design





PWM – Code

💿 pwm | Arduino 1.8.7 (Windows Store 1.8.15.0)

File Edit Sketch Tools Help





Demo 4 – Servo Control



Servo Motors







Servo Motors

- Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire.
- The PWM sent to the motor determines position of the shaft, and based on the duration of the pulse.
- The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns.
- For example, a 1.5ms pulse will make the motor turn to the 90° position. Shorter than 1.5ms moves it in the counter clockwise direction toward the 0° position, and any longer than 1.5ms will turn the servo in a clockwise direction toward the 180° position.





Servo Control – Circuit Design





Servo Control – Code

servo | Arduino 1.8.7 (Windows Store 1.8.15.0)

File Edit Sketch Tools Help



Servo myservo;

// twelve servo objects can be created on most boards

int pos = 0; // variable to store the servo position

```
void setup() {
```

myservo.attach(9); // attaches the servo on pin 9 to the servo object

1

```
void loop() {
```



Demo 5 – Analog Input



Analog Input

- Reads the analog value from the specified analog pin.
- Arduino maps input voltages between 0 and 5 volts into integer values between 0 and 1023. (10 bit ADC)
- This yields a resolution between readings of: 5 volts / 1024 units or, .0049 volts (4.9 mV) per unit.
- It takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.


Analog Input – Potentiometer – Circuit Design





Analog Input – Potentiometer – Code

analog | Arduino 1.8.7 (Windows Store 1.8.15.0)

File Edit Sketch Tools Help

	© COM8
analog	0 0
int potPin =3;	320 938
<pre>int potValue = 0;</pre>	1023 755
<pre>void setup () { Serial.begin(9600);</pre>	755 169 789
}	0
<pre>void loop() {</pre>	0 0
<pre>potvalue = analogRead(potPin); Serial.println(potValue);</pre>	Autoscroll Show timestamp
delay(500);	
}	





