

<u>https://1.bp.blogspot.com/-</u> <u>XhQVLp4bA34/TzAnlNdwZHI/AAAAAABCbI/NCj49D2uP8w/s1600/Pierre+Auguste+Renoir+Flowers+-</u> <u>+Tutt%2527Art%2540+%252817%2529.jpg</u>

Physics 524 Survey of Instrumentation and Laboratory Techniques 2023

George Gollin University of Illinois at Urbana-Champaign

Unit 1a: Arduino C++ programming environment

Week 1: Organizations, Distributions, and Installations

Goal for this week	2
Notebooks	2
Distribution of stuff!	2
Installing the Arduino programming IDE; running code	3
In-class exercise: blink code	9
Breadboarding!	9
In-class exercise/milestone 1b: BME680	11
In-class (due at the first class meeting next week)	15
How to read a schematic	15
This week's homework assignment (due at the first class meeting next week)	17

Goal for this week

• You will assemble a starter circuit on a breadboard and use the Arduino Integrated Developers Environment (IDE) to program and test it.

Notebooks

Please open up your notebook, and add to your account of the afternoon's activities as you work. You'll want to keep track of your work, and your revelations, so you can return to them at a later date. We won't be looking at your notes, but if I ask you something like "where did you go to find that cool piece of demo software?" you should be able to give me an answer based on your notes.

Distribution of stuff!

I have an alarmingly large amount of stuff available for you. Some of the following I've already packed in the parts/tools kits I'll distribute in class. (You'll also use these in Physics 523.)

part
16 x 2 LCD CFAH1602B-NGG-JTV
2 x AAA & 1 x AAA battery holders
3 x 4 keypad (#3845)
5 x AA battery holder, Adafruit 3456
74HC137 3-8 decoder/demultiplexer (TTL)
74HC157N quad 2 input (TTL) multiplexer
Adalogger Feather M0, Adafruit 2796, and microUSB – USB A cable
ADXL326 accelerometer (#1018)
ALLPOWERS 2.5W, 5V photovoltaic (solar) cell

Arduino Mega 2560 and USB cable
BME 680 T/RH/P/VOC (#3660)
Diodes: 1N5817 Schottky
DS3231 real time clock (#3013)
electret microphone with amplifier (#1063)
RadioFeather M0 with 900 MHz LoRa radio (#3178)
GPS Antenna - External Active Antenna (#960)
INA219 battery voltage/current sensor (#904)
LSM9DS1 accelerometer/magnetometer/gyroscope 0x1E (#3387)
MCP23008 I2C port expander, Adafruit 593
MCP4725 DAC, 12-bit, I2C (#935)
MicroSD card breakout (#254)
MicroSD memory card 8 GB SDHC (#1294)
Mini Metal Speaker w/ Wires - 8 ohm 0.5W (#1890)
MLX90614 IR sensor. Note that there are distinct 3V and 5V versions!
PAM8302 Audio Amplifier (#2130)
red LEDs (#297)
TCA9548A mux I2C multiplexer
TMP36 analog medium temperature thermometers (#165)
TSL2561 or TSL2591 Digital Luminosity/Lux/Light sensor
ultimate GPS breakout board (#746)
USB DIY connectors

Installing the Arduino programming IDE; running code

Go to the Arduino webpage <u>https://www.arduino.cc/en/software</u>. Download and install the Arduino Desktop IDE (Integrated Development Environment) on your laptop.

The heart of your initial explorations will be an Arduino Mega 2560 microcontroller board, shown here.



Arduino Mega 2560

It's a remarkable little gizmo, featuring an Atmel Atmega2560 microcontroller (built by Microchip Technology, Inc.) running at 16MHz. The Atmega2560 has 256kB of flash memory in which your program will reside, along with 8kB of SRAM (static random access memory) in which will live the variables your program modifies as it executes. There are 16 analog inputs that feed an internal multiplexer whose output drives a 10-bit successive approximation analog to digital converter (ADC). See https://store.arduino.cc/arduino-mega-2560-rev3 for more details.

Some of the projects might involve a different processor: I have been developing systems using several Adafruit devices, two of which employ Microchip ATSAMD21G18 microcontrollers. The SAMD21 runs at 48 MHz, features a 12-bit ADC, and has 32 kB of SRAM, four times more than the Atmega2560. It also has a 10-bit digital to analog converter (DAC), which the 2560 does not.

The Adalogger M0 includes a built-in microSD memory holder, while the RadioFeather M0 with RFM95 LoRa radio has a built-in 915 MHz long range radio transceiver. Both are very cool.

The Arduino IDE is quite a bit simpler than Anaconda's iPython IDE. Most of what you will see on your screen is an editor window in which you will create/modify C++ programs that you will compile and upload to the Arduino. See the screen shot, below. You'll write and compile programs using the IDE, then upload them to the Arduino through a USB cable.

There isn't a debugger, so you'll be forced to print things to a "serial monitor" screen to keep track of what's going on (and going wrong) in the code executed by the Arduino.

Something to keep in mind: the Arduino runs code that is very much like C++, with some minor differences. But the structure of a program is constrained: there are always two specific functions that must be included in a program. The first is called "setup"; it takes no arguments and does not return a value. It is the first routine in a program that executes. The next is called "loop." It too takes no arguments and does not return a value. Upon completion of setup, the

Arduino goes straight into the loop function. It executes loop over and over, jumping back into it each time the function completes.

Do this: plug the Arduino into a USB port on your laptop, then fire up the IDE. Open the Blink example, compile and upload it, and see if your processor will talk to you. You'll need to make sure the IDE knows what kind of processor you are using: a Mega 2560. (Select this from the pull-down menu which, in my screen shot, below, initially says "Adafruit Feather M0 (SAMD21)."

Here are some screen shots:

🗯 Arc	duino IDE	File Edit Sketch Tools	Help	
		New Sketch % N	Built-in examples	C Dich L Archine IDE 0.44
-	•	New Cloud Sketch T # N	01.Basics	AnalogReadSerial
		Open #O	02.Digital	BareMinimum
		Open Recent >	03.Analog	Blink
	Blink.in	Sketchbook >	04.Communication	DigitalReadSerial
	1	Examples	05 Control	Fade
-	, 2	Close 98W	06 Sensors	ReadAnalogVoltage
Ľ-	3	Save % S	07.Display	
	4	Save As	08 Strings	у.
	5		09USB	
	6	Most Arduinos have	10 StarterKit BasicKit	GA and ZERO
	> 7	it is attached to c	11 ArduinoISP	is set to
10	8	the correct LED pir		
	9	IT you want to know	Examples for Adafruit Feather M0 (SAMD21)	our Arduino
Q	10	model, check the le	Adafruit TinyUSB Library	
	11	https://www.arduine	CI_Tests >	
	12	modified 8 May 2014	Ethernet >	
	14	by Scott Eitzgerals	Firmata >	
	15	modified 2 Sep 2016	I2S >	
	16	by Arturo Guadalupi	Keyboard >	
	17	modified 8 Sep 2016	LiquidCrystal >	
	18	by Colby Newman	SAMD_AnalogCorrection >	
	19		SDU >	
	20	This example code i	Servo >	
	21		SPI >	
	22	https://www.arduinc	Stepper >	
	23	*/	TFT	
	24		USBHost >	
	25	<pre>// the setup functior</pre>	Wire >	ard
	26	<pre>void setup() {</pre>	Examples from Custom Libraries	
	27	<pre>// initialize digit</pre>	Accelerometer H3LIS331DL	
	28	pinMode(LED_BUILTIN	Adafruit ADXL343	
	29	}	Adafruit ADXL375	
	30	// the leas function	Adafruit AS726X	
	31	// the toop function	Adafruit AS7341	
	32	digitalWrite(LED PL	Adafruit BME680 Library	voltage
	22	level)	Adafruit BusIO >	voltage
	34	delay(1000):	Adafruit DPS310	
	35	digitalWrite(LED BL	Adafruit EPD >	e voltage
		LOW	Adafruit FRAM I2C >	
	36	delay(1000);	Adafruit GFX Library	
0	37	}	Adafruit GPS Library	
8	38		Adafruit HX8357 Library	
			Adafruit ILI9341	
			Adafruit ImageReader Library	
			Adafruit INA219	
			Adafruit Keypad	
			Adafruit LC709203F	
			Adafruit LED Backpack Library	
			Adafruit LIS331	

Opening the "blink" example.

_					
• •	•			🔤 Blink Ardı	lino IDE 2.1.1
	€ €	Adafruit Feather M	(SAMD21) -		
	Blink.ino		`		
_	1	/*	<u>د</u>		
57	2	Blink			
	3				
D-0.	4	Turns a	on for one second, then off fo	or one secon	d, repeatedly.
ШЛ	5	March Andreas	have an exchanged LED over any		the UNIO MECA and ZEDO
	5	Most Arduinos	nave an on-board LED you can	control. Un	The UNU, MEGA and ZERU
	/	the correct l	I to digital pin 13, on MKRI00	on pin o.	LED_BUILTIN IS SET to
	9	Tf you want t	Select Other Board and Port		×
\bigcirc	10	model check			
~	11	https://www.a	Select both a Board and a Port if you	want to upload a	sketch.
	12		IT you only select a Board you will be a	able to compile, b	out not to upload your sketch.
	13	modified 8 Ma	BOARDS		POBTS
	14	by Scott Fitz			
	15	modified 2 Se	mega	Q	
	16	by Arturo Gua			
	17	modified 8 Se	Arduino Mega ADK		/dev/cu.BLTH Serial Port
	18	by Colby Newm	4 1 1 14 14 0500		
	19		Arduino Mega or Mega 2560	~	/dev/cu.blue leethBuds Serial Port
	20	This example	Arduino NG or older		/dev/cu.Bluetooth-Incoming-Port Serial Port
	21				
	22	https://www.a	Arduino Nano Every		/dev/cu.usbserial-142320 Serial Port (USB)
	23	*/	Arduino Uno WiFi Rev2		
	24	// the setup for			
	25	void setun() {	Atmel atmega168pb Xplained mini		
	27	// initialize			□ Show all ports
	28	pinMode(LED E			
	29	}			
	30				CRITCEE
	31	// the loop fun			
	32	<pre>void loop() {</pre>			
	33	digitalWrite(_ED_BUILTIN, HIGH); // turn t	the LED on (I	HIGH is the voltage
		level)			
	34	delay(1000);	// wait f	for a second	
	35	digitalWrite(FD BIITITTN. IOW): // turn t	the LED off	by making the voltage

Identifying the processor to be used.

You'll need to tell the IDE which port to use:



Specifying the USB port to use.

What the program does is pretty obvious. Good coding references are on the Arduino site: see <u>https://www.arduino.cc/reference/en/</u> and <u>https://www.arduino.cc/en/reference/libraries</u>.

You can compile the program upload it to your Arduino by clicking the right-arrow button near the top of the window:



Note the presence of the setup and loop functions.

Some more technical commentary: many of the Arduino's "pins" are configurable: they can be defined to be digital inputs, or outputs, or analog inputs. The pinMode instruction defines the pin driving the red LED to be a (digital) output.

Unit 1a

Here's what you can do to view program output in a "serial monitor" window. First add some code to write to it:

```
*/
23
24
25
    // the setup function runs once when you press reset or power the board
    void setup() {
26
27
       // initialize digital pin LED_BUILTIN as an output.
28
       pinMode(LED_BUILTIN, OUTPUT);
29
30
       // light up the serial monitor and wait for it to be ready, but include a
31
       // timeout in case there are problems.
       // define a 32 bit, unsigned integer and load it with the number of
32
33
       // milliseconds since the program began executing.
34
       uint32_t t1234 = millis();
35
36
       // Light up the serial mnonitor at 115,200 baud (bits per second)
37
       Serial.begin(115200);
38
       // wait for the serial line to be ready: it'll return "true" when it's up.
39
       while (!Serial && millis() - t1234 < 5000) {/* do nothing */}</pre>
40
41
       // let's assume everything went OK, so we'll write stuff to the
42
43
       // serial monitor.
       Serial.println("Blink program is starting up!");
44
45
     }
46
47
     // the loop function runs over and over again forever
48
    void loop() {
       // turn the LED on (HIGH is the voltage level)
49
50
       digitalWrite(LED_BUILTIN, HIGH);
51
       // let the user know
52
       Serial.println("LED on");
53
       // wait for a second
       delay(1000);
54
55
       // turn the LED off
       digitalWrite(LED_BUILTIN, LOW);
56
57
       Serial.println("LED off");
       // wait for a second
58
59
       delay(1000);
60
```

Then set the baud rate:

	Ln 44 Col 41 - Arduino Maga or Maga 2560 on /dev/cu usbserial-142320 🤇 2 - 🗖
)	:50:35.196 -> LED on
	:50:34.181 -> LED off
	:50:33.175 -> LED on
	:50:32.162 -> LED off
	:50:31.147 -> LED on
	:50:30.136 -> LED off
	:50:29.159 -> LED on
	:50:28.153 -> LED off
	:50:27.138 -> LED on
	:50:26.120 -> LED off
	:50:25.104 -> LED on
	:50:24.104 -> LED off
	:50:23.088 -> LED on
	:50:22.077 -> LED off
	essage (Enter to send message to 'Arduino Mega or Mega 2560' on '/dev/cu.ust New Line • 115200 baud •

In-class exercise: blink code

Modify the blink program so that it blinks the initials (of your English/American name) in Morse code.

Breadboarding!

Let's prep our breadboards. Please fasten an Arduino to your breadboard. I recommend duct tape (the baby sitter's friend!); position the device to that it doesn't cover any of the breadboard's plastic structures that are used to hold components.

Here's how the holes are interconnected, underneath the plastic surface. The five holes in a column are connected, as shown in a few spots in the picture below. The 25 holes in a horizontal row are also connected.

In the photo after the close up I show a breadboard with an Arduino and connections between the Arduino's 5V and grounds to the breadboard. The doubled connection of the Arduino grounds is not electrically necessary, but it does provide redundancy in case one of the ground wires falls out.

Be sensible in your choices of wire colors: always use red for 5V and black for ground. You'll want to strip about 5 mm of insulation from each end of a wire when establishing your connections.

		30	g	7				g		E = E
C							 			3+
	3			5	40 * * *		 50	55		
		30	9	2	50	g L	 01	ç	0	A
C		Ē								+



In-class exercise/milestone 1b: BME680

Most of our breakout boards were built by Adafruit Industries, a wonderful provider of small electronic packages intended largely for the hobbyist market. Go to the Adafruit site <u>https://www.adafruit.com/</u> and find the BME680 page that mentions some of the supporting infrastructure available for you.

Install the BME680 onto your breadboard. (See <u>https://www.adafruit.com/product/3660</u> and links therein.) You should power it using the Arduino's +5V and GND lines. Using sensible colors (red for +5V, black for ground, other colors for signal lines), connect GND to one of the Arduino's GND lines and VIN to one of the Arduino's 5V lines. Also connect the leads of a 0.1μ F capacitor to the BME680's power and ground inputs.



We'll let the device and the Arduino communicate using an I2C ("I Two C": Inter Integrated Circuit) interface; set this up by connecting the BME680's SCK (serial clock) pin to the Arduino's SCL output (pin 21). Also connect the BME680's SDI (serial data) to the Arduino's SDA input (pin 20). You should leave unconnected the BME680's 3Vo, SDO, and CS pins.

Here's a schematic for the circuit that also includes a liquid crystal display. We'll discuss briefly how to read the schematic. (Don't bother with the LCD for the moment, though you are free to wire it up, if you'd like.)

ш Ο ∢ contrast-setting potentiometer P398DLP_TRIMPOT_DEV p524_BME680_LCD 8/2/23 10:37 AM 1/1 Sheet: LCD USE George Golfin University of Illinois at Urbana-Champaign GND VO RN RN 9 BME680 T/P/RH/VOC စုံစုံနှ စ္စစ္စစ္စရ My pin numbering conventions: physical breakout board pins are numbered from left to right schematic symbol pins are numbered from top to bottom ഹ S LCD DB4 LCD DB5 LCD DB6 LCD DB6 LCD_RS LCD_E GND Physics 524 Simple test circuit ^тС_ВМЕ680 Place LCD with pins at TOP. Left-most pin is 1, right-most is 16. 0.1uF ∃u£.0 I2C_SDA GND I2C_SCL crco GND 52 GND 4 LCD DB4 LCD DB5 LCD DB6 LCD DB6 *********** GND power Arduino MEGA 2560 Lica Component side is DOWN. stacr D141 D156 D156 D161 D161 D181 D187 D187 D187 D187 D187 D20S Ľ งการ Parts you'll need: Arduino MEGA 2560 (1) BME680 (1) 0.1 uF capacitors (3) LCD display (1) 10k trim potentiometer (1) D1< power jack 28 BATTERY BATTERY LCD_E GND СИD ٨S ∢ ш C

8/2/23 10:37 AM /Users/g-gollin/Super/Woodchuckie2/Physics_education/professional_masters/Eagle_MEng/projects/p524_BME680_LCD.sch (Sheet: 1/1)

You'll need to install one of Adafruit's libraries to drive the BME680. See https://learn.adafruit.com/adafruit-bme680-humidity-temperature-barometic-pressure-voc-gas/arduino-wiring-test and scroll down to the section titled "Install Adafruit_BME680 library." Follow the directions to install the library and upload to the Arduino the demonstration software. Then open, compile, and run the example program BME680test.

🗯 Ard	luino IDE	File Edit Sketch Tools	Help	
		New Sketch % N	Built-in examples	
• •	•	New Cloud Sketch てまい	01.Basics	> Arduino IDE 2.1.1
		Open XO	02.Digital	>
		Open Recent >	03.Analog	>
P	Blink_202	Sketchbook >	04.Communication	>
	1	Examples >	05.Control	>
_	2	Close %W	06.Sensors	>
1_)	3	Save % S	07.Display	>
	4	Save As 쇼울 S	08.Strings	>
līk	5		09.USB	>
	6	Most Arduinos have a	10.StarterKit_BasicKit	> and ZER0
	7	it is attached to di	11.ArduinoISP	> .s set to
12	8	the correct LED pin	Examples for Arduino Mega or Mega 2560	
\sim	9	If you want to know	FEDROM	Ir Arduino
Q	10	model, check the lec	Ethernet	
	11	nttps://www.arduino.	Firmata	
	12	modified 8 May 2014	Keyboard	× .
	13	hy Scott Eitzgerald	LiquidCrystal	
	14	modified 2 Sep 2016	Servo	
	16	by Arturo Guadaluni	SoftwareSerial	× .
	17	modified 8 Sep 2016	SDI	
	18	by Colby Newman	Stepper	
	19		TET	
	20	This example code is	Wire	× .
	21			· · · · · · · · · · · · · · · · · · ·
	22	https://www.arduino.	Examples from Custom Libraries	
	23	*/	Accelerometer_H3LIS331DL	>
	24		Adafruit ADXL343	>
	25	<pre>// the setup function</pre>	Adafruit ADXL375	> .q
	26	<pre>void setup() {</pre>	Adafruit AS726X	>
	27	<pre>// initialize digita</pre>	Adafruit AS7341	>
	28	<pre>pinMode(LED_BUILTIN,</pre>	Adafruit BME680 Library	> bme680async
	29	-	Adafruit BusIO	> bme680oled
	Output		Adafruit DPS310	> bme680test
	Sketch	n uses 2794 bytes (1%) (Adafruit EPD	> s.
	Global	l variables use 232 byte	Adafruit FRAM I2C	> r local variables. Maximum is 8192 bytes.
	avrduo	<pre>le: stk500v2_ReceiveMes:</pre>	Adafruit GFX Library	>
	avrdude: stk500v2_ReceiveMes: avrdude: stk500v2_ReceiveMes:		Adafruit GPS Library	>
			Adafruit HX8357 Library	>
	avruut	le: stk500v2_ReceiveMes:	Adafruit ILI9341	>
	avrduo	le: stk500v2_ReceiveMest	Adatruit ImageReader Library	>
	avrduo	de: stk500v2 getsvnc():	Adatruit INA219	
	Failed	d uploading: uploading (Adafruit Keypad	
			Adatruit LC709203F	>
			Adatruit LED Backpack Library	>
			Adatruit LISSOT	
			Adatuit LISSNUL	Vpload error: Failed uploadin
			Adamut LSM9D50 Library	
\bigcirc			Adamut Lowedon Library	>
8			Adamuit MCD22008 library	
			Adamut MCP25000 library	Ln 5, Col 1 Arduino Mega c
fn ^ N	#		Adafruit MCD/728	
fn ctrl opt	ion cmd		Audit uit NGF4/20	

/dev/cu.usbmodem301 Send BME680 test Temperature = 29.04 *C Pressure = 988.0500 hPa Humidity = 62.30 % Gas = 0.00 KOhms Approx. Altitude = 211.95 m Temperature = 29.13 *C Pressure = 988.0500 hPaHumidity = 62.37 % Gas = 249.01 KOhms Approx. Altitude = 211.95 m Temperature = 29.26 *C Pressure = 988.0700 hPa Humidity = 61.75 % Gas = 260.02 KOhms Approx. Altitude = 211.61 m Temperature = 29.32 *C Carriage return 9600 baud Clear output Autoscroll Show timestamp

You should find that the pressure transducer is so sensitive that it can tell that you've lifted the board up from your worktable by a couple of feet just from the change in atmospheric pressure.

In-class (due at the first class meeting next week)

On your breadboard, install the following devices (in addition to the BME680 and Arduino): LCD (including $10k\Omega$ trimpot), keypad, and microSD breakout. See the schematic, below.

For each device find a demo program (perhaps on the Adafruit site, or from a library that you might install, or from the course's "Code & design resources repository") and confirm that it functions properly. You'll need to fool around with the 10k trimpot to adjust the LCD contrast properly.

How to read a schematic

A schematic holds a topological representation of an electronic circuit. The two most important things on it are symbols for the various components—resistors, capacitors, integrated circuits and so forth—and (named) nets, which define the electrical connections between components. For example, the BME680 symbol on the schematic shows seven pins, with pins 1, 3, 4, and 6 connected to the nets named 5V, GND, I2C_SCL, and I2C_SDA, respectively. (Pins 2, 5, and 7 are not connected to anything.) Anything tied to the 5V net is electrically connected to everything else on the 5V net.



When lines representing nets cross each other the point of crossing does *not* represent a connection between the nets! For example, the LCD_RS and LCD_E nets are not connected to each other here:



We indicate a point at which two lines (which are on the *same* net) are electrically connected with a dot as follows:



This week's homework assignment (due at the first class meeting next week)

Finish all of the in-class exercises in this unit, and be prepared to show your results to the course staff during class.

