CS2610 Homework Assignment 2

Getting Physical and Creating Internet of Things (IoT) with Arduino and Galileo

Release date: 9/30/2015   Due Date: by 9:00AM on 10/19/2015

This homework assignment should be completed in three/four-member groups. Only one copy of report is needed for each group. Please contact the TA if you couldn't find a partner for this assignment. Late submissions will loss 20% of the total grade each day. We won't accept submissions later than 3 days.

The students may freely choose their desired programming language for this assignment.

Background

"*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*." Mark Weiser envisioned the future of computing in his seminal paper in 1991 [1]. Indeed, as researchers in computer science, the computers "on our radar" shouldn't only be PCs sitting on our desktop – machines in large data centers, cell phones/tablets hiding in people's pockets or bags, and those intelligent sensors embedded in our daily environments are all important research topics in this "post-PC" era. The motivation of this homework assignment is to let you be aware of technologies and opportunities on building interfaces and interactive systems beyond traditional desktop GUIs via a set of exercises designed for Arduino [2] and Intel Galileo [4] (figure 1).

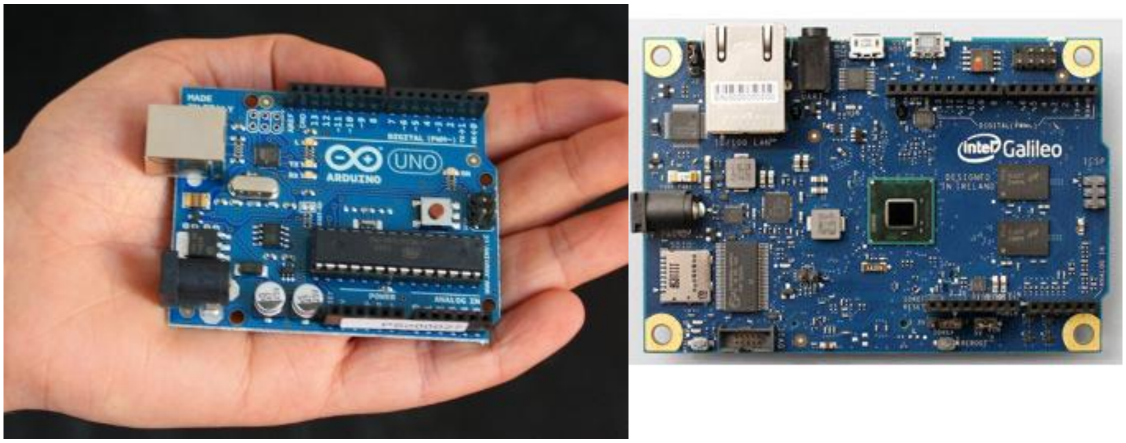

Figure 1. The Arduino UNO Board (left)[2] and the Intel Galileo Board (right) [4]

Arduino is an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The hardware consists of a simple open hardware design for the Arduino board with an 8bit Atmel micro-controller and on-board I/O support. The software consists of a standard programming language and the boot loader that runs on the board [2]. The Arduino hardware is programmed using a Wiring-based language (syntax + libraries), similar to C++ with some simplifications and modifications, and a Processing-based IDE. Both the hardware and the software tools support all major OSes such as Windows, Mac OS X, and Linux for development purposes.

The Arduino hardware has gone through multiple rounds of design revisions and upgrades since its first release in 2005. We will use its latest version, code named "Arduino UNO" (figure 1) for our homework assignment this time. To give you some basic ideas, Arduino UNO has an ATmega328 processor, 32 KB flash (for saving programs), 2 KB RAM, 14 Digital I/O pins and 7 analog I/O pins.

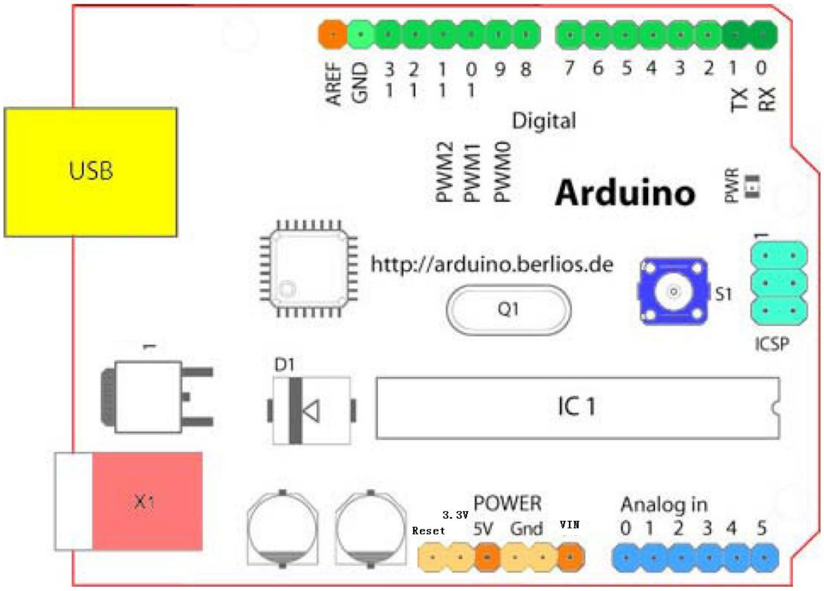

Figure 2. Major components of an Arduino Board, Please refer to [3] for detailed explanations.

The hardware of the Arduino is a board containing an ATmega microcontroller (IC1 in Figure 2) surrounded by several digital and analog I/O pins. The board also has a USB port which can power-up the board and exchange data when connected to a host computer. The Arduino board can also be powered up by 9V DC input (X1 in figure 2) when necessary. There are two major usage scenarios of Arduino. First, Arduino can be used as an interface board to send information to the host computer and generate physical activities via lights/sounds/actuators by following the host computer's instructions (Figure 3.a). Second, Arduino can also be used as a standard alone, highly portable computer without the help of a host computer. In the second mode, The program should be able to fit into the 32 KB on board flash memory and 9V power (X1 in Figure 2) via either a battery or an AC Adapter is necessary (Figure 3.b). Do the descriptions above sound like a digital circuit lab you experienced during your undergraduate study? No, the major focus of this assignment is not about wiring electronic components;

it's about sensing and interacting with physical world via software programs in a way that is not possible by pure desktop GUI programming.

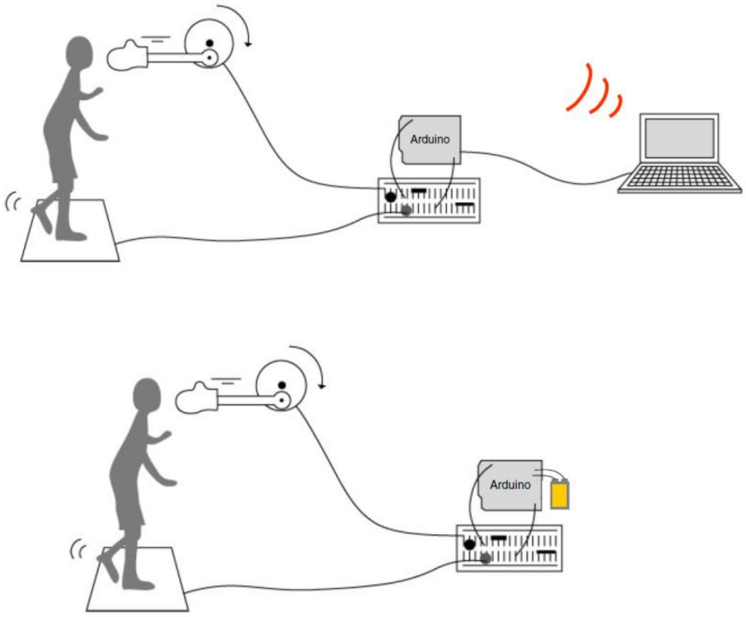

Figure 3. The two usage scenarios of Arduino (From top to bottom: **a.** As an interface board; **b.** as an embedded computing device)

The Intel Galileo Gen 2 is a powerful platform designed for prototyping high performance physical computing, embedded computing, internet of things (IoT), and wearable computing systems. Intel Galileo are compatible with Arduino [5] in the high-level programming language, the IDE, and the on-board pins. More importantly, Galileo is an Intel x86 compatible, Pentium class System-on-a-Chip (SoC), with a 400 MHZ CPU and 256MB Ram (the original Arduino board only has 32KB Ram). Because of this, Galileo is capable of running a full-strength OS (Windows, Linux), high-level programming languages such as Python, and computation intensive tasks such as real-time image processing. In this assignment, you may simply treat the Intel Galileo Gen 2 board as a faster Arduino board or explore advanced features by installing and running a Linux OS [5][6] in it.

Components Lent to You

Please contact the TA to get the following required components for this assignment. **The members need to sign a "receipt of equipment" to get the components and the members should return the components in good and working condition after finishing the homework assignment.** Before we get started, let's first have a reality check. Here are the items you will receive.

- One Arduiono UNO board (figure 1, left)
- One Intel Galileo board (figure 1, right)

- One Mini Bread Board
- A set of jumper wires (if you believe the jumper wires are not enough, please get more from local electronics stores such as RadioShack, or online stores such as sparkfun.com or ebay.com)
- One Mini Push Button (single throw) (figure 4.a [8])
- One 10 Segment LED Bar Graph (figure 4.b [9])
- One Triple Output LED (figure 4.c [10])
- Three single color LEDs (Red, Green, Yellow, one for each)
- One Mini photocell (figure 4.d [11])
- One Mini Buzzer (figure 4. e [12])
- One Piezo Element (generating sound or detecting vibration) (figure 4.f)
- **Two** force sensitive resisters (figure 4.g [13])
- One Thermistor 10K (figure 4.h [14])
- One HC-SR04 ultra sound distance sensor (figure 4.i [15])
- One ESP8266 Serial-to-WiFi Module (figure 4.j [16])

Please read the corresponding descriptions in the links provide before start using them!

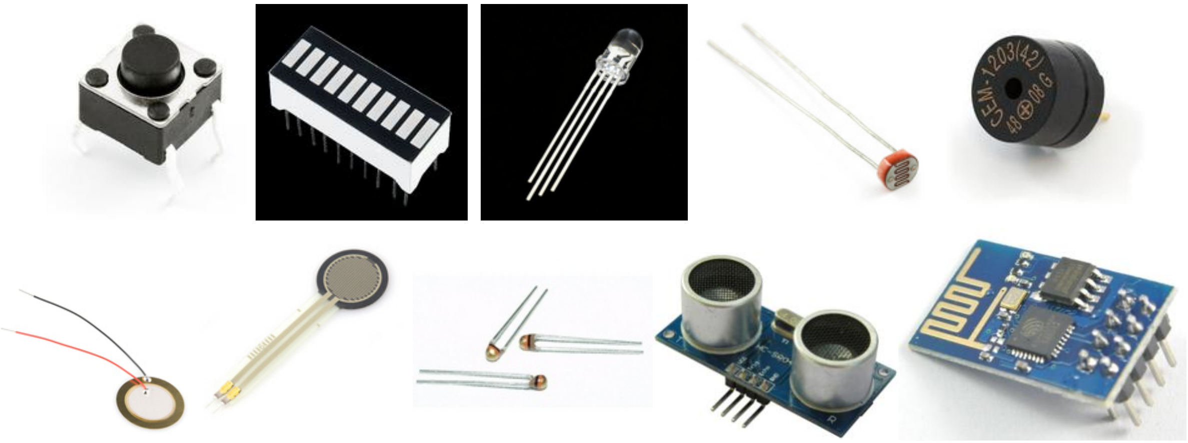

Figure 4. Some components for this homework assignment (From left to right, top to bottom, **a**. Mini Push Button [8], **b**. 10 Segment LED [9], **c**. Triple Output LED [10] **d**. Mini photocell [11] **e**. Mini Buzzer [12] **f**. Piezo Element **g**. Force sensitive resistor [13] **h**. Thermistor [14] **i.** Distance sensor [15] **j**. Serial-to-wifi module [16])

Each group can also get some resistors from the TA for the homework assignment. Most likely you will need resistors marked with (220 ohm, 10K okm and 1M ohm). **Due to the size limit of the current course, each group can get no more than two resistors of the same capacity**. Please refer to [7] on how to read resister color codes. If you need more than two resisters of the same value for your intended design, please get them from RadioShack by yourself. Most likely you will need more than two 220 ohm resisters to drive LEDs (for restricting current purposes), but you can use resistors of similar values (e.g. 120 ohm, 150 ohm, 330 ohm) to replace them.

The Item You Need to Prepare

Considering the wide availability of USB powered devices (printers, external hard drives etc), each group need to get a standard USB cable (A-plug to B-plug, figure 5) for this homework. Please note that our Arduino UNO board does not accept USB cables with mini/micro B-plug (e.g. those used by some smart phones). If your group wants to do something more creative, please feel free to get components not provided in this class for the homework assignment.

Figure 5. A Standard USB Cable (A-plug to B-plug)

Assignments

Note: Steps 0 – 4 need to be completed by every team member independently. Team members should work together for question 5 and 6.

0. **Setting up the Arduino Software and connect it to an Arduino UNO board.** Please follow instructions provided in [17]. Make sure you read the environment section and check the trouble shooting section. You should be able to run the blink example if you completed everything successfully. (No need to connect an additional LED to the board for this example. There is a mini on-board LED marked “L” connected to digital pin 13 and the blink sample code will drive that LED). Make sure you can understand the wiring language and the example, try to change the parameters of the example (e.g. blinking duration, or use another pin to drive the LED, in this case you need to attach an additional LED to the GND pin and the digital pin [19]) and reload it to Arduino. Read the

Arduino Foundations [18] web page to make you more familiar with the Arduino hardware and software library. Start playing with some examples in the tutorial section [20]. If you are using an Intel Galileo, you need to download the Intel customized Arduino IDE from Intel's website [6].

1. **Blinking Multiple LEDs automatically (5 Points).** In this assignment, you need to drive multiple LEDs (e.g. three LEDs, by using either one triple output LED, three LEDs or the 10 segment LED bar). You need to turn on each LED for 500 milli-seconds and then it turn-off and turn on the next LED.

2. **Driving Multiple LEDs with a push button(5 Points).** In this assignment, you need to use one push button to control multiple LEDs. Please use three LEDs to represent a 3-bit digit counter, pressing the button each time will add the internal state of the counter by one and hence change the state of the three LEDs.

3. **Two-key musical keyboard (5 points).** Implement a two-key musical keyboard by following the three-key musical keyboard example [19].

4. **Knock Sensor (5 Points).** Please use the Piezo element provided and an LED to implement a knock sensor. Knocking your table once will turn on the LED, knocking it again will turn-off the LED. Please refer to [22] for inspirations.

5. **Serial communication (10 points).** Connect two force sensors and one photocell to Arduino, read values from these sensors and send them to the PC. From the PC side, write a program to visualize these sensor values (the easiest visualization is displaying both the numeric value and a corresponding bar chart for each channel). There are a few examples in the communication section of Arduino tutorial [20] for you to refer to. These examples are written in processing, a language good at create images, animations, and interactions. If you want to use a different language (such as Java, C/C++, Python, Flash, Perl, Ruby etc), please refer to [23] for details.

6. **Creative Ideas (20 points + up to 3 bonus points).** Please use the components provided creatively to build something that interacts with the physical world. The requirement is – a) your team should use some unique input/output capabilities enabled by Arduino; b) functions provided by your system should require the help of programs running on the PC/server side. Some examples include – using the two force sensors to play the "BreakOut" game on PC; Driving LEDs/Buzzers by sending different tweets to your twitter account (think about controlling/monitoring your coffee machine or microwave at home from your office); Getting access to a desktop program you created by knocking a set of secret codes on the table; reading the values of certain sensors (e.g. light, temperature) and send them to a server or desktop program and visualize them; Grabbing real time bus information from a web site (can be a mock-up website) and use LEDs or sounds to indicate upcoming buses. It will be better if you can leverage some expertise from one of your current research directions. **Please don't be too aggressive in this question; we are not expecting a class project quality submission from your group.** It's okay to use libraries and components on the internet to speed up your development, but you need to acknowledge them explicitly in your report. We hope you can use this chance to brainstorm the opportunities enabled by sensing and interacting with the physical

world. **Your team will receive extra credit if you could make the Android device "wireless" by taking advantage of the ESP8266 wifi module (Figure 4.j) and the Microsoft Azure server**.

Some Closing Words

Due to time and resource constraints, this assignment only scratched the surface of technology for interacting with the physical world. Arduino is a highly extensible platform – communication modules for wifi (XBee), Bluetooth, Ethernet, and 3G can be added easily to Arduino and then Arduino can act as a node in a much bigger wireless sensor network; Other off-the-shelf sensors such as current, capacitive, proximity, motion, biometric, bar-code, etc. can be easily attached to Arduino due to its highly flexible digital/analog I/O ports and programming model; analog/step motors, servos and power relays can be used to enable richer output. Arduino has been widely used in both industry and academia, for both research [24] [25] and product development [26]. By working on this assignment, I hope you will have an opportunity to "connect the dots" some day.

What to Submit

1. Upload some small video clips (no more than 20 seconds each for assignments 1 – 4; no more than 1 minute for A5; no more than 2 minutes for A6) of your completed system to the homework assignment wiki page. Please use voice-overs and/or on-screen captions to explain your system. No need to use a camcorder if you don't have one, the video recording function in your digital camera or camera phone will be fine.

2. Email a brief report (no more than 3 pages in total) that describes your solutions to both the instructor and the TA. Describe challenges you met and lessons learned during the implementation. A couple of sentences will be okay for assignments 1 – 4. Use figures when necessary; make sure your descriptions are accurate and concise.

3. Email a copy of the compressed source code of your project to the TA (please also include a README file to describe how to build your application. Please also include **a web link** to a compressed copy of pre-built, ready to run application on the PC side in the README file). Please make sure the total size of your source code package is no more than 1.5 MB. Please contact the instructor and the TA if you couldn't make the source of your project smaller than the size limit.

References

[1] Weiser, M. The Computer for the 21th Century. Scientific American, September, 1991.

[2] Arduino, http://www.arduino.cc

[3] Arduino UNO Mainboard, http://arduino.cc/en/Main/ArduinoBoardUno

[4] Intel Galileo http://arduino.cc/en/ArduinoCertified/IntelGalileo

[5] Intel Galileo Get Started https://communities.intel.com/docs/DOC-22872

[6] Intel Galileo Linux and Arduino IDE Download https://communities.intel.com/docs/DOC-22226

[7] How to Read Resistor Color Codes http://wiki.xtronics.com/index.php/Resistor_Codes

[8] Mini Push Button http://www.sparkfun.com/datasheets/Components/General/00097.jpg

[9] 10 Segment LED Bar Graph (and youtube demo) http://www.sparkfun.com/products/9935

[10] Triple Output LED http://www.sparkfun.com/products/105

[11] Mini Photocell http://www.sparkfun.com/products/9088

[12] Mini Buzzer http://www.sparkfun.com/products/7950

[13] Force sensitive resister http://www.sparkfun.com/products/9375

[14] Thermistor 10K http://www.sparkfun.com/products/250

[15] Ultrasound distance sensor http://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/

[16] ESP8266 Serial-to-wifi module
http://rancidbacon.com/files/kiwicon8/ESP8266_WiFi_Module_Quick_Start_Guide_v_1.0.4.pdf
and http://fab.cba.mit.edu/classes/863.14/tutorials/Programming/serialwifi.html

[17] Getting started with Arduino http://arduino.cc/en/Guide/HomePage

[18] Arduino Foundations http://arduino.cc/en/Tutorial/Foundations

[19] Blink Example http://arduino.cc/en/Tutorial/Blink

[20] Arduino Examples http://arduino.cc/en/Tutorial/HomePage

[21] Musical Keyboard http://arduino.cc/en/Tutorial/Tone3

[22] Knock Sensor http://arduino.cc/en/Tutorial/Knock

[23] Interfacing with Arduino http://www.arduino.cc/playground/Main/InterfacingWithSoftware

[24] Jon E. Froehlich, Eric Larson, et al, HydroSense: Infrastructure-Mediated Single-Point Sensing of Whole-Home Water Activity, In Proc of Ubicomp 2009

[25] Greg Niemeyer, Antero Garcia, Reza Naima, Black cloud: patterns towards da future, In Proc. of ACM MultiMedia 2009

[26] AR-Drone, http://ardrone.parrot.com/parrot-ar-drone/usa/