Development of Smart Linear Velocity Measuring Device by Embedding Sensors with the Arduino Microcontroller

Vincent A. Balogun
Department of Mechanical Engineering, Faculty of Engineering, Edo University Iyamho, Edo State, Nigeria
vincent.balogun@edouniversity.edu.ng

Omonigho B. Otanocha
Department of Mechanical Engineering, College of Technology, Federal University of Petroleum Resources, P.M.B 1221, Effurun, Delta State, Nigeria
otanocha.omonigho@fupre.edu.ng

Bankole I. Oladapo
College of Engineering, Department of Mechanical & Mechatronics Engineering, Afe Babalola University, Ado Ekiti, Ekiti State, Nigeria
bankolyable@yahoo.com

ABSTRACT

The change of an object's position with respect to time and a reference point is usually adopted to determine its velocity. This is usually a function of time. Velocity can be deduced in terms of its speed and direction of motion. It is an important concept in the description of kinematics and mechanics of bodies which entails the definitions of the bodies magnitude and direction. This project is one of the teaching series of smart technology development for the step change and technological awareness on the introduction of the IoT global influx. It focuses on the measurement of velocity using ultrasonic sensor. The Ultrasonic module HC - SR04 that has a range between 2 cm - 400 cm and a non-contact measurement function was adopted and linked to the Arduino UNO board for data processing and conversion before the output displays the specific measurement of the object velocity. The accuracy of the Ultrasonic module HC - SR04 ranges up to 3 mm. This module comprises of the ultrasonic transmitters, the receiver and the control circuit [1]. This is to teach and encourage students within the developing countries such as Nigeria to exploit and contribute to the global trend of the IoT smart technology advancement.

KEYWORDS

Ultrasonic sensor, Arduino, IoT, Instrumentation, smart technology

IML '17, October 17-18, 2017, Liverpool, United Kingdom

ACM ISBN 978-1-4503-5243-7/17/10…$15.00
https://doi.org/10.1145/3109761.3158399

1 INTRODUCTION

Velocity is a vector quantity[1] that consists of a magnitude (speed) and a direction. The linear velocity of any particle can be defined as the rate of change of the distance with time [2]. There are several instruments found in literature that are used to measure the linear velocity of a solid object. For example 1) the Linear velocity transducer (LVT), that is an inductive device, it utilizes the link between electricity and magnetism [3] as proposed by Lorentz [2, 4]. 2) the Doppler radar velocity measurement [5, 6]; this device adopts the principles whereby the frequency of a noise that is moving has an apparently higher (or lower) pitch or frequency depending on the route [7]. This is so because the wavelength of the sound that reaches the ears (in this case the receiver) is compacted (or stretched) as a result of the relative motion. However, this principle works with radio waves for Doppler radar. The frequency of the radio waves reflected is altered in a similar manner as the sound waves when the radio waves strike an object that is moving 3) Velocity measurement using displacement and acceleration sensors [2, 8, 9].

It is a common practice for the velocity of particle to be evaluated from the measurement of displacement or acceleration rather than been measured directly [10]. The displacement (distance travelled) can be evaluated with a displacement sensor [11]. By fundamental definition, velocity V, is the time derivative of displacement x, as shown in equation 1, thus,

$$V(t) = \frac{dx(t)}{dt} \tag{1}$$

Hence, the differentiation of distance with respect to time is usually obtained from the displacement sensor for example the potentiometer, LVDT, laser displacement meter, etc [2]. However, it is possible for an inherent error of multiplication to exist with this technique of theoretical differentiation and evaluation. This is so because the differential manipulation of the measured signal will as well amplifies the noise within the system thereby decreases the signal-to-noise ratio. Thus, this method of velocity measurement by differentiation of displacement data is generally not the best idea as the noise within the system is also magnified and differentiated alongside the required signal. However, if the displacement sensor has an extremely high signal-to-noise ratio, then the method can be adopted since the noise within the system will be negligible at that instance.

Also, a photo detector, one of the many types of an optical sensors, are frequently adopted to show the time that a moving object passes through the sensor's optical path [2]. Hence it becomes very easy to evaluate the average speed of the object when two of such sensors are located at a known distance apart as shown in equation 2. The velocity of very fast-moving objects can be measured with this method.

$$V_{avg} = \frac{\Delta x}{\Delta t} \tag{2}$$

where x and t represents displacement in *mm* and time in *s* respectively.

The acceleration sensors could also be adopted in some instruments, it measures acceleration as a function of time. By fundamental definition, velocity is the time integral of acceleration as shown in equation 3.

$$V(t) = V_0 + \int a(t)dt \tag{3}$$

where V_0 is the velocity at time t_0. The integration process, if adopted, can decrease the noise in the system by increasing the signal to-noise ratio. It is therefore recommended that the velocity measurement by integration of the acceleration variables should generally be adopted [2].

The ultrasonic sensors vibrates at some ultrasonic frequencies as a result of an acoustic transducer embedded in them. The mode of operation is such that the target object receives the beam pulses that are emitted from the cone-shaped sensors. Usually, pulses reflected by the target to the sensor are detected as echoes [12]. The time delay between each of the signal emitted and the reflected echo pulse is measured by the device. This strategy enables the device to accurately determine the distance to target required by the device. This device can be adopted to detect materials that are very clear, translucent, dark and opaque and shiny targets. The sensors also work very well in harsh environments (i.e. fumes, dust, noisy, etc). They are available in different sizes and ranges. For example the HC - SR04 provides 2 cm - 400 cm non-contact measurement function [13]. The ranging accuracy of the ultrasonic sensors can reach up to 3 mm and are usually presented in modules that includes ultrasonic transmitters, receiver and control circuit [13].

1.2 The Arduino Microcontroller applications

The Arduino is a form of integrated circuit embedded and a simple microcontroller board with an open source development environment that allows computers to drive both functional and creative projects [14]. The Arduino hardware with an open hardware design consists of the Atmel AVR processor [13]. The pre-assembled Arduino boards are commercially available and can be purchased locally. It is also possible to build one from scratch with the hardware design information.

Several add-on boards (shields) produced by third-parties are able to broaden the basic functionalities of an Arduino. The Motor Control Shield is a very good example of the add-on boards that allows users to read encoders and to control DC motors. The Xbee shield, (allows wireless communication between several Arduino boards) and the Critical Velocity Accelerometer Shield (can integrate a 3-axis accelerometer) [13] are good examples of add-on boards.

The Arduino shown in Figure 1 has software that consists of a standard programming language and a firmware that runs on the board. However, it is possible that non-experts can adopt and use the Arduino boards without any programming experience. This is one critical advantage of the open source philosophy of the Arduino boards (both hardware and software).

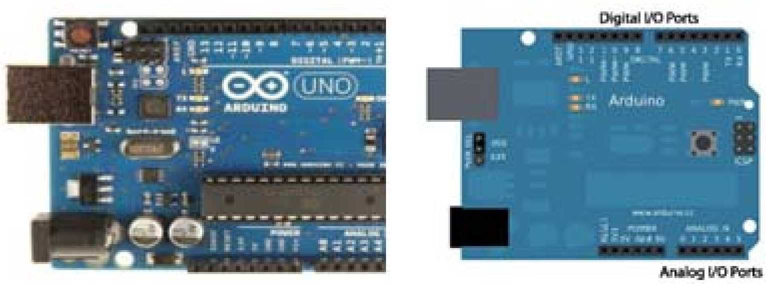

Figure 1: The Arduino UNO board showing the external appearance and the simplified schematic of the input/output ports.

The software uploaded into the Arduino hardware can be programmed with a simplified C++ language in a processing-based integrated development environment (IDE). Arduino boards are compatible also with other hardwares like Flash, MaxMSP, and MATLAB [13]. The arduino structure for basic programming is composed of the setup and the loop components. The set-up parts states and defines all variables in order to set the pin mode or the serial communication. The loop components enables the coded script to interrelate through the changing, responding, and controlling of the Arduino board [13].

After all variables have been declared, other controlling functions of the Arduino are then set. The controlling functions involves classic control statements and/or structures (i.e. IF, IF…ELSE, FOR, etc.), arithmetic operators (+, -, /, *, etc.), and comparison operators (>, <, etc.) or Boolean (AND, OR, etc.). There is also a set of commands for analog and digital read and write such as *digitalwrite()* or *digitalread()* controlling functions [13, 15, 16].

The electric current supply of the Arduino is limited to about 40 mA through any one of its output pins. To use the Arduino to control a high-powered device requires the use of an "amplifier" or "signal-buffer". An amplifier simply reproduces a low-power input signal, with a much higher output power to drive a load [17, 18]. A basic amplifier has an input and an output. The input is a low-power signal (like the Arduino) and is used to drive the larger output signal that will power the load. A perfect amplifier is of the type that can easily switch the high-power signal very quickly and efficiently in the same manner as the Arduino switches the low-power signal [18].

It has been reported [13] that the possibility to load the experimental script on the board's memory to run without interfacing with any computer or external software is one of the major strengths of Arduinos. This action grants complete independence, portability, and accuracy to the Arduinos [19]. Furthermore, free scripts for different projects on Arduino are available in literature [13]. Hence making the incorporation and integration of Arduino into any project easier and straightforward.

The Arduino has been applied to many IoT based research. For example, in smart irrigation systems, for torque measurement systems [20, 21], for characterizing the multi-layered matrix textile sensor in the development of an interface for pressure measurement [22], for smart monitoring of Air Borne PM 2. 5 density level [23], also for the development of a laboratory model for automated road defect detection [24], for the development of smart glove system for therapy treatment [25], etc.

1.3 Aim and Objectives

The aim of this work is to develop an smart velocity measuring device by embedding sensors with the Arduino Microcontroller. It focuses on the measurement of velocity using ultrasonic sensor ranging module HC - SR04 which could provide 2 cm - 400 cm contactless measurement function. The specific objectives are to:

- determine appropriate velocity sensor to be integrated with the Arduino microcontroller
- simulate the functionability of the assembly
- develop the prototype and conduct the performance evaluation of the prototype.

2 THE STRATEGY FOR DEVELOPMENT AND RESULTS ANALYSIS

2.1 The Arduino Board

The adopted Arduino Uno is a microcontroller board based on the ATmega328. The Arduino has 14 digital input/output pins. Six of the pins can be used for the Pulse Width Modulation (PWM) outputs [13, 19]. It also has analog inputs 1) crystal oscillator of 16-MHz, 2) USB connection, 3) power jack and 4) serial header programmed in-circuit. A USB connection is used to power. the Arduino Uno Alternatively, the Arduino Uno can be powered with an external power supply from a 9-V battery [13]. This work adopts the strategy of the Arduino powered via a USB connection. it is important to note that the board operates on an external supply of 6 to 20 volts. The ATmega328 has 32 KB (with 0.5 KB used for the bootloader) and also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library) [26].

Each of the 14 digital pins on the Uno can be used as an input or output, using *pinMode(), digitalWrite(),* and *digitalRead()* functions [26]. They operate at 5 volts. Each of the pin can provide or accept a maximum of 40 mA, and has an internal pull-up resistor of 20–50 $k\Omega$ [26]. In addition, some of the pins have specialized functions. For example Pins 0 and 1 may be used to receive (RX) and transmit (TX) transistor-transistor logic (TTL) serial data; Pins 2 and 3 can be configured to trigger an interrupt of a low or change in value, or falling edge [13]. Finally, Pins 3, 5, 6, 9, 10, and 11 provide 8-bit Pulse Width Modulation (PWM) output with the *analogWrite()* function. The Arduino Uno has six analog inputs, labeled A0 through A5, each of which provides 10 bits of resolution (1,024 different values). By default, they measure from ground to 5 volts [13].

2.2 Hardware and Software for the Velocity Sensor

To produce the velocity sensor, the following hardware were employed; Arduino Uno, USB cable for the Arduino, 16 x 2 Liquid Crystalline Display (LCD), 9 volts' battery, HC-SR04 ultrasonic sensor, 380 ohms' resistor and jumper wires. The Software includes Arduino sketch and Fritzing.

This project was designed and simulated with the Fritzing software platform. Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone [20, 21, 27]. Fritzing is a software tool, for creative symbiotic system used to document prototypes, for sharing and teaching electronics and for layout and manufacture of professional PCBs.

After the design and simulation on the Fritzing platform, the pins were connected to their respective spots as shown in Figures 2 and 3. The HC-SR04 sensor plugins is connected as follows: the Trig pin was connected to the digital port 2, Echo pin to digital port 4, the 5 v to the VCC port and GND to GND port.

The LCD display plugins is also connected as follows: PIN 1 or VSS connected to the ground, the PIN 2 or VDD or VCC to +5 v power, PIN 3 or VEE to ground (gives maximum contrast best for a beginner), PIN 4 or RS (Register Selection) is connected to PIN 8 of the ARDUINO UNO, PIN 5 or RW (Read/Write) to ground (puts LCD in read mode and eases the communication for user), PIN 6 or E (Enable) connected to PIN 9 of the ARDUINO UNO, PIN 11 or D4 to PIN 10 of the ARDUINO UNO, PIN 12 or D5 linked to PIN 11 of the ARDUINO UNO, PIN 13 or D6 connected to PIN 12 of ARDUINO UNO, and PIN 14 or D7 connected to PIN 13 of the ARDUINO UNO [28].

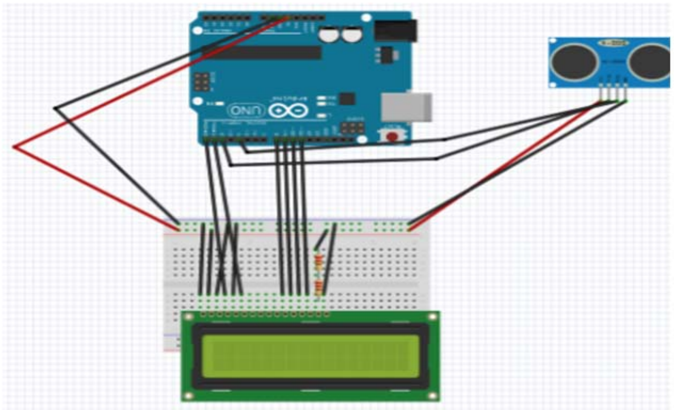

Figure 2: Connection diagram for the device.

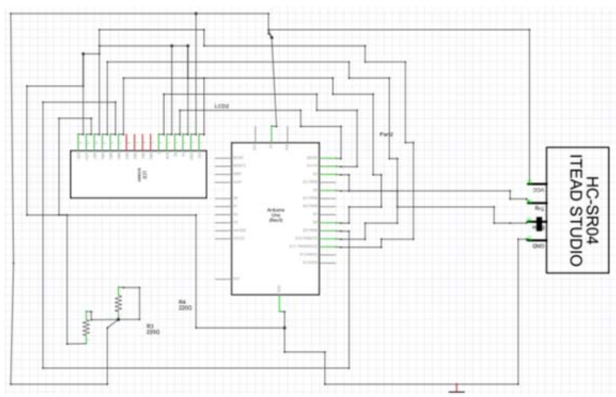

Figure 3: Schematic diagram of the device.

The programming commenced as soon as the wires were completely connected. The Arduino IDE was utilized to upload the code to the board.

Figure 4 shows the wiring script that was deployed to the Arduino Uno used in all the tests.


```
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(0, 1, 8, 9, 10, 11); /// REGISTER SELECT PIN,ENABLE PIN,D4 PIN,D5 PIN, D6 PIN, D7 PIN
const int trigPin = 2;
const int echoPin = 4;
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
          // Print a logo message to the LCD.

                lcd.print("OLANOYE, ");
                  lcd.setCursor(0,1);
                   lcd.print("UTHMAN OLADIPO");
                    delay (2500);
                    lcd.clear();//clear display;

                 lcd.begin(16, 2);
                 lcd.print("SADIQ, ");
                  lcd.setCursor(0,1);
                   lcd.print("GEGELE MUBARAK");
                    delay (2500);
                    lcd.clear();//clear display

                   lcd.print("AYORINDE, ");
                  lcd.setCursor(0,1);
                   lcd.print("TEMITOYIN");
                    delay (2500);
                    lcd.clear();//clear display


}
long distanceOverTime(long first,long second){
 return ((first-second)/.1)*.0223693629;//taking cm/s to mph
}

long holder;//store the cm from last time through loop.
long temp;//used to store the speed value after changes


void loop() {
  long duration, inches, cm;
  long Speed;
  // The sensor is triggered by a HIGH pulse of 10 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(trigPin, OUTPUT);

void loop() {
  long duration, inches, cm;
  long Speed;
  // The sensor is triggered by a HIGH pulse of 10 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(trigPin, OUTPUT);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);

  // convert the time into a distance
  cm = microsecondsToCentimeters(duration);

  Speed=distanceOverTime(holder,cm);
  holder=cm;//after speed caclulation so take the cm value for another calculation
  Speed=abs(Speed);
  delay(250);
  lcd.clear();
  lcd.print(cm);
  lcd.print(" cm ");
  lcd.print(Speed);
  lcd.print(" mph ");
  lcd.println();
}
long microsecondsToCentimeters(long microseconds)
{

  return microseconds / 29 / 2;
}
```

Figure 4: The wiring script.

3 RESULTS AND CONCLUSIONS

3.1 The working principle of the developed device.

The linear velocity of an object using the non-contact techniques was adopted to develop the linear velocity measurement device. This techniques adopts the sensing capabilities of the ultrasonic sensors for measurements. Although it is important to consider the maximum velocity, maximum measurement range, accuracy, and operating temperature of the linear velocity sensors deployed for the device, these parameters were not considered in this project. This is because the project is to demonstrate the possibility of deploying the Arduino Uno in the measurement of linear velocity for teaching and learning. These parameters would be considered in advancing the development further.

In this design, as the solid moving object moves closer or further away from the ultrasonic sensor attached to the device, the distance travelled by the object is automatically determined and the time of approach is also determined. The final computation and evaluation is done within the script as shown in Figure 4 and the result which is displayed on LCD output reveals the linear velocity of the travelling object. Data can however be exported in ASCII format and thereafter loaded in MATLAB software in order to carry out further analyses on them. Tests can also be designed to demonstrate and verify the timing accuracy with increasingly complex movement tasks [13]. This further proves that the Arduino UNO is an accurate platform for a series of lab settings.

Due to the oblique area of projection of the signal sent out by the ultrasonic sensor, it is difficult for it to signal out on particular object and tell its speed. One of the factors required before choosing a measuring instrument is the sensitivity and the environment in which its application would be employed, so the perfect environment to use the ultrasonic would be in an isolated system. The preferred solution to this problem would be to use the LDR5 light sensor which utilized a laser and only reacts when an obstacle breaks the laser. Therefore as long as no obstacle crosses the laser, no unnecessary interference would occur with the readings.

4 CONCLUSIONS

This study has shown that linear velocity measuring device by embedding sensors can be demonstrated with the Arduino Microcontroller Arduino Uno. This strategy can also be employed for other measuring devices most especially for integrated learning resources of the applications of IoT technology in developing countries. Also through this type of measurement equipment design, it is possible that the fidelity of velocity measurement is maintained and hence reliable decisions on speed limitations can be made for various moving objects. Therefore, a smart sensing module can ensure the robustness and integrity of measurement devices if and when adopted with them.

REFERENCES

[1] SainSmart. (2010, 4/5). *Ultrasonic Ranging Detector Mod HC-SR04 Distance Sensor*.
[2] J. M. Cimbala, "Linear Velocity Measurement," *Penn State University,* 2013.
[3] D. Jiles, *Introduction to magnetism and magnetic materials*. CRC press, 2015.
[4] H. A. Lorentz, *The Einstein Theory of Relativity: A Concise Statement*. Brentano's, 1920.
[5] O. A. Phillips, "Doppler radar velocity measurement horn," ed: Google Patents, 1987.
[6] O. A. Phillips, "Doppler radar velocity measurement apparatus," ed: Google Patents, 1988.
[7] S. Widodo *et al.*, "Moving object localization using sound-based positioning system with doppler shift compensation," *Robotics,* vol. 2, no. 2, pp. 36-53, 2013.
[8] F. Yi *et al.*, "Self-Powered Trajectory, Velocity, and Acceleration Tracking of a Moving Object/Body using a Triboelectric Sensor," *Advanced Functional Materials,* vol. 24, no. 47, pp. 7488-7494, 2014.
[9] Z. L. Wang, L. Lin, J. Chen, S. Niu, and Y. Zi, "Self-powered Sensing for Tracking Moving Objects," in *Triboelectric Nanogenerators*: Springer, 2016, pp. 455-467.
[10] M. C. Varley, I. H. Fairweather, and Aughey1, Robert J, "Validity and reliability of GPS for measuring instantaneous velocity during acceleration, deceleration, and constant motion," *Journal of sports sciences,* vol. 30, no. 2, pp. 121-127, 2012.
[11] J. G. Webster and H. Eren, *Measurement, instrumentation, and sensors handbook: spatial, mechanical, thermal, and radiation measurement*. CRC press, 2014.
[12] K. Audenaert, H. Peremans, Y. Kawahara, and J. Van Campenhout, "Accurate ranging of multiple objects using ultrasonic sensors," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, 1992, pp. 1733-1738: IEEE.
[13] A. D'Ausilio, "Arduino: A low-cost multipurpose lab equipment," *Behavior research methods,* vol. 44, no. 2, pp. 305-313, 2012.
[14] W. Durfee, "Arduino microcontroller guide," *Course Material, University of Minnesota,* pp. 1-27, 2011.
[15] S. F. Barrett, "Arduino microcontroller processing for everyone!," *Synthesis Lectures on Digital Circuits and Systems,* vol. 8, no. 4, pp. 1-513, 2013.
[16] M. Evans, J. J. Noble, and J. Hochenbaum, *Arduino in action*. Manning, 2013.
[17] M. Banzi and M. Shiloh, *Getting started with Arduino: the open source electronics prototyping platform*. Maker Media, Inc., 2014.
[18] J.-D. Warren, J. Adams, and H. Molle, "Arduino for robotics," in *Arduino robotics*: Springer, 2011, pp. 51-82.
[19] S. Ferdoush and X. Li, "Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications," *Procedia Computer Science,* vol. 34, pp. 103-110, 2014.
[20] A. Knörig, R. Wettach, and J. Cohen, "Fritzing: a tool for advancing electronic prototyping for designers," in *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, 2009, pp. 351-358: ACM.
[21] Z. Bogdanovic, K. Simic, M. Milutinovic, B. Radenkovic, and M. Despotovic-Zrakic, "A Platform for Learning Internet of Things," *International Association for Development of the Information Society,* 2014.
[22] I. Baldoli, M. Maselli, F. Cecchi, and C. Laschi, "Development and characterization of a multilayer matrix textile sensor for interface pressure measurements," *Smart Materials and Structures,* vol. 26, no. 10, p. 104011, 2017.
[23] P. P. Ray, "Internet of Things Cloud based smart monitoring of Air Borne PM2. 5 density level," in *Signal Processing, Communication, Power and Embedded System (SCOPES), 2016 International Conference on*, 2016, pp. 995-999: IEEE.
[24] H. Bello-Salau, A. Aibinu, E. Onwuka, J. Dukiya, A. Onumanyi, and A. Ighabon, "Development of a Laboratory Model for Automated Road Defect Detection," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* vol. 8, no. 9, pp. 97-101, 2016.
[25] A. M. M. Ali, Z. M. Yusof, A. Kushairy, and A. Ismail, "Development of smart glove system for therapy treatment," in *BioSignal Analysis, Processing and Systems (ICBAPS), 2015 International Conference on*, 2015, pp. 67-71: IEEE.
[26] C. Components. (2018, 14/6). *Everything Arduino & Genuino*.
[27] Banana-soft. (2017, 14/7). *Fritzing*.
[28] Arduino. (2018, 14/7). *What is Arduino?* Available: https://www.arduino.cc/en/Guide/Introduction