New Trends and Issues Proceedings on Humanities and Social Sciences

Volume 6, Issue 6 (2019) 058-071

www.prosoc.eu

Selected Paper of 6th Global Conference on Contemporary Issues in Education 29-31 August 2019, St. Petersburg, Russia

Design of smart stick for visually impaired people using Arduino

Ramiz Salama*, Department of Computer Engineering, Near East University, Nicosia, Cyprus
Ahmad Ayoub, Department of Computer Engineering, Near East University, Nicosia, Cyprus

Suggested Citation:
Salama, R. & Ayoub, A. (2019). Design of Smart Stick for Visually Impaired People Using Arduino. *New Trends and Issues Proceedings on Humanities and Social Sciences.* [Online]. *6*(6), pp 058–071. Available from: www.prosoc.eu

Selection and peer review under responsibility of Prof. Dr. Huseyin Uzunboylu, Near East University, North Cyprus

Abstract

Nowadays, blind or impaired people are facing a lot of problems in their daily life since it is not easy for them to move, which is very dangerous. There are about 37 million visually impaired people across the globe according to the World Health Organization. People with these problems mostly depend on others, for example, a friend, or their trained dog while moving outside. Thus, we were motivated to develop a smart stick to solve this problem. The smart stick, integrated with an ultrasonic sensor, buzzer and vibrator, can detect obstacles in the path of the blind people. The buzzer and vibration motor are activated when any obstacle is detected to alert the blind person. This work proposes a low-cost ultrasonic smart blind stick for blind people so that they can move from one place to another in an easy, safe and independent manner. The system was designed and programmed using C language.

Keywords: Arduino Uno, Arduino IDE, ultrasonic sensor, buzzer, motor.

* **ADDRESS FOR CORRESPONDENCE: Ramiz Salama,** Department of Computer Engineering, Near East University, Nicosia, Cyprus. *E-mail address:* ramiz.salama@neu.edu.tr / Tel.: + 90 533 841 81 42

1. Introduction

People with visual impairment are the individuals who find it hard with healthy eyes to acknowledge the smallest detail. Individuals with the visual acuity of 6/60 on the horizontal range with both eyes open are considered to be blind. A study conducted by the WHO (World Health Organization) in 2011 estimated that around 1% of the world's population is visually impaired (about 70 million individuals), with around 10% of them being completely blind. The main problem with blind people is their navigation (Grum & Grum, 2017).

Visually impaired people are the people who find it difficult to recognize the smallest detail with healthy eyes (Mostafa & Mta'en, 2017). Those who have visual acuteness of 6/60 or whose horizontal range of visual field with both eyes open is less than or equal to 20 degrees are regarded as blind. A survey by the WHO in 2011 estimates that, around the world, about 1% of the human population is visually impaired (about 70 million people) and among them, about 10% are fully blind (about 7 million people) and 90% (about 63 million people) have low vision. Blind people need assistance from others with good eyesight. As described by the WHO, 10% of the visually impaired have no functional eyesight and they cannot safely move around without assistance. Figure 1 shows a chart illustrating the blind people present across the globe. This study proposes a new technique for designing a smart stick to help visually impaired people to navigate safely. The conventional and archaic navigation aids are the walking cane (also called white cane or stick) and guide dogs, which are characterized by many imperfections. The most critical shortcomings of these conventional aids include essential skills, training phase, range of motion and very insignificant information communication. Our approach modified the conventional cane with some electronics components and sensors. The electronic aiding devices are designed to solve the above-mentioned issues. The ultrasonic sensors, water sensor, buzzer and RF transmitter/receiver are used to record information about the presence of obstacles on the road. The ultrasonic sensor can detect any obstacle within the distance range of 2–450 cm. Therefore, whenever there is an obstacle in this range, it will alert the user. The water sensor is used to detect water in the path of the user. Most blind guidance systems use ultrasound because of its immunity towards environmental noise. With the rapid advances of modern technology both in hardware and software, it easy to provide an intelligent navigation system to the visually impaired. Recently, much research effort has been focused on the design of Electronic Travel Aids to aid successful and free navigation and high-end technological solutions have been introduced to help blind people navigate independently. Another reason why ultrasonic technology is prevalent is due to its cost-effectiveness. Moreover, ultrasound emitters and detectors are portable components that can be manufactured without using complex circuits. The RF module will help the person to find the stick wherever it is placed. Whenever the user wants to locate the stick, a button on the remote control has to be pressed, which will activate the buzzer to ring. Vision is the most important part of human physiology since 83% of the information that a human being gets from the environment is via sight. The 2011 statistics by the WHO estimates that there are 70 million people around the world living with visual impairment, of which 7 million are blind and 63 million have low vision. The conventional and oldest mobility aids for persons with visual impairments are characterized by many limitations. Some inventions also require a separate power supply or a navigator which has to be carried in a bag every time during outdoor travel. These bulky designs will definitely make the user to be exhausted.

Moving through an unknown environment becomes a real challenge when we can't rely on our own eyes. Since dynamic obstacles usually produce noise while moving, blind people develop their sense of hearing to localize them. A white cane is the most common mobility aid for the visually challenged. However, it does not give information about the obstacles above knee level and those which are at a distance greater than 1m. Even though guide dogs were the initial companion of the blind, later on, technologies played a vital role. Walking sticks with adjustable length, e.g., elbow canes, were developed and sold in the market to guide the visually challenged. However, these attempts were not completely successful in assisting the user. To alleviate these issues, the smart electronic aid is designed in such a way that it includes an ultrasonic sensor for obstacle detection, supported with

heat and water detection. In this system, vibratory motors are used to inform about the moving obstacles. The intensity of vibration depends on the speed of the moving obstacles. Vision is the most important part of human physiology as 83% of information human being gets from the environment is via sight. The 2011 statistics by the WHO estimates that there are 285 billion people around the world with visual impairment, 39 billion of them are blind and 246 have low vision. The traditional and oldest mobility aids for persons with visual impairments are the walking cane (also called Smart Stick) and guide dogs. The most important drawbacks of this stick are necessary skills and training phase, range of motion and very little information conveyed. With the rapid advances in modern technology, both in hardware and software front have brought potential to provide detection of obstacles. Walking safely and confidently without any human assistance in urban or unknown environments is a difficult task for blind people. Visually impaired people generally use either the typical white cane or the guide dog to travel independently.

Although the white stick gives a warning about 1 m before the obstacle, for a normal walking speed of 1.2 m/second, the time to react is very short (only 1 second). The stick scans the floor and consequently cannot detect certain obstacles (rears of trucks, low branches, etc.). Safety and confidence could be increased using devices that give a signal to find the direction of an obstacle-free path in unfamiliar or change in environments. Smart sticks are devices that give off a warning by buzzer or/and vibrator when an obstacle is in the way and allow the user to avoid the obstacles.

2. Objective

The primary objective is to help people with visual impairment to navigate easily and independently to lead their own private and independent life. This stick came as a partial solution to this problem because, on the street, home, schools and university, people with these disabilities always need help from others. And unfortunately, in most countries, because of their disabilities, these people feel incomplete in society. The intelligent walking stick, built with the highest degree of precision, will assist the blind individuals to move from one location to another without anybody's assistance.

3. Literature survey

This article on Sensor Assisted Stick for the blind proposes a light flexible stick which is connected with an ultrasonic sensor to scan the atmosphere surrounding the blind individual to send data to the buzzer and an engine to inform the user of a hazard or barrier to avoid accidents. The primary element for this system's operation is the ultrasonic sensor that is used by emitting reflecting waves to scan a predetermined region around the blind individual. As inputs to the Arduino microcontroller, the reflected signals are obtained from the objects. The microcontroller is then used to determine the around objects' direction and distance.

4. Proposed work

The ultrasonic sensor functions like human eyes in the suggested scheme. The ultrasonic sensor sends ultrasonic waves and the waves repulse back. The sensor detects the barrier and the distance between the blind individual and the barriers as a result of this procedure and sends data to the microcontroller. The sound-based ultrasonic sensors operate. The sound waves are transmitted forward from the sensors to the obstacle which, with a resolution of 0.3 cm, can feel the distance up to 12 feet. The sensors are put in five places to cover the highest possible sides with minimal sensor use. The sensors are left, right, centre left, centre left and centre left.

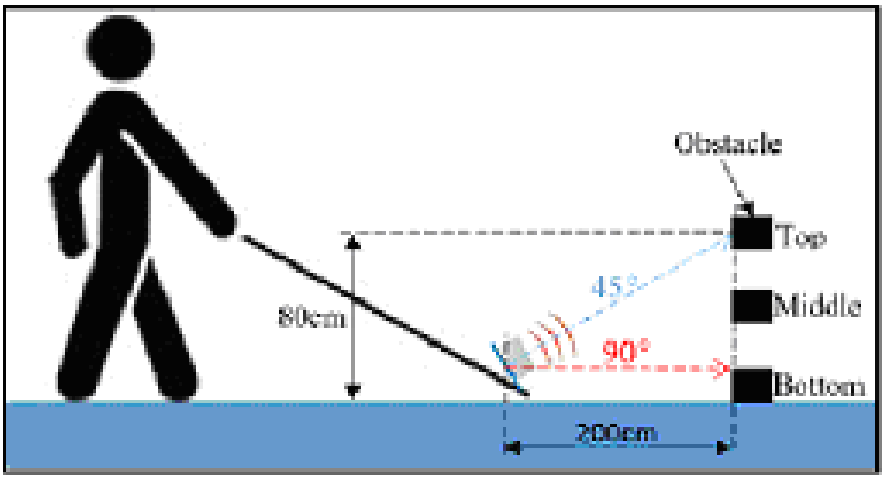

Figure 1. Working of blind stick

5. Block diagram

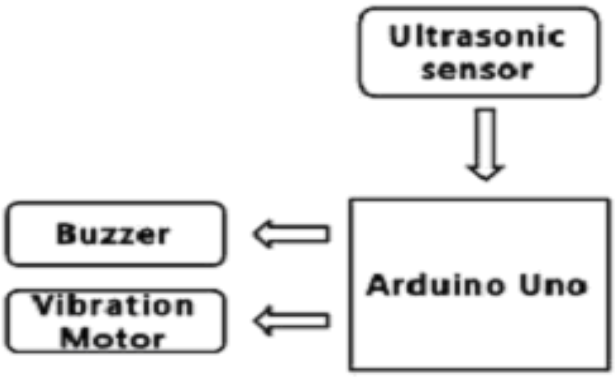

Figure 2. Block diagram of the project

6. Explanations of components

This scheme is used to identify the barrier by the ultrasonic sensors (if any). The sensors are set to identify barriers within the range of variance chosen by the individual. Obstacles discovered in various directions are stated for easy identification with distinct beep patterns. The ultrasonic sensors emit sound scopes that are inaudible to humans with ultrasonic spectrum frequencies (>20 kHz).

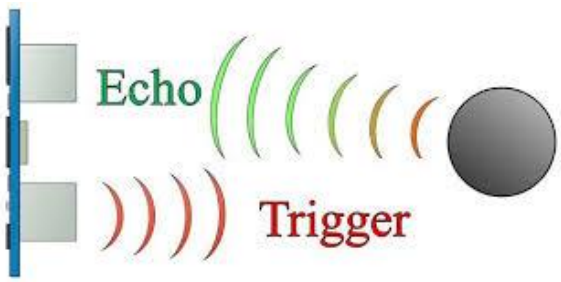

Figure 3. Explanation of the ultrasonic sensor mechanism

7. Working principle

The system's primary benefit is that it helps blind individuals to realise carefree navigation in indoor and outdoor environments. The tools in the stick are simple and comfortable to manage. The intelligent stick helps to detect barriers in front of the user at a specific distance. The system is appropriate for indoor as well as outdoor environments. The data about barriers are provided through the buzzer, eliminating the trouble of knowing the patterns of vibration that were used in previous schemes. This stick is a cost-effective portable navigation aid for the visually impaired.

The system's primary portion is the microcontroller that regulates the other system parts.

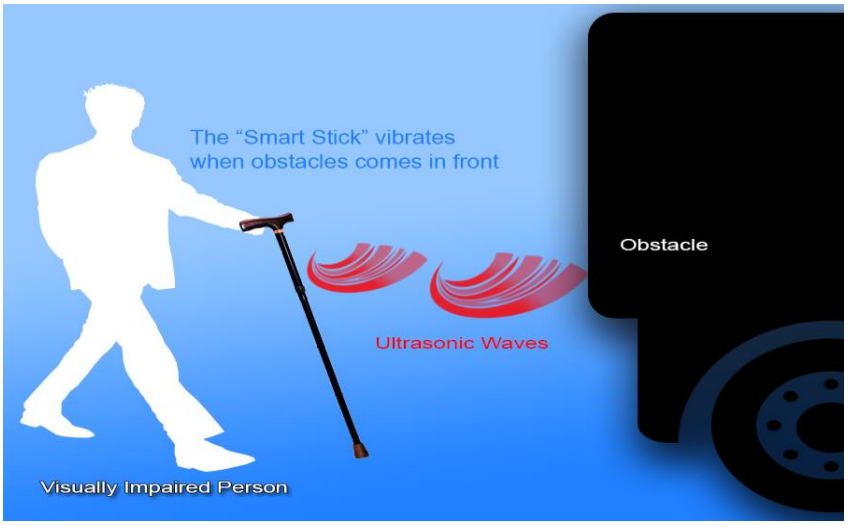

Figure 4. Working principle of smart blind stick

8. Software requirements

The Arduino Uno is a simple yet strong single-board computer that has gained significant momentum in the hobby and professional sectors. The Arduino is an open source, meaning the hardware is priced fairly and the software free to develop. This guide is for confrontation with the Arduino students for the first time. Arduino IDE is open source software that is mainly used for writing and compiling the code into the Arduino Module.

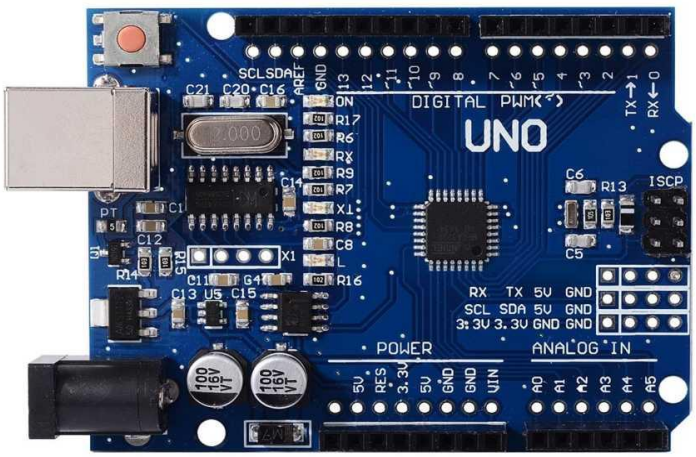

Figure 5. Arduino Uno

9. The code and software used

What is Arduino? Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards read inputs—light on a sensor, a finger on a button or a Twitter message—and turn it into an output—activating a motor, turning on an LED and publishing something online. The board can be instructed by sending a set of instructions to the microcontroller on the board through Arduino programming language (based on wiring) and Arduino Software (IDE; based on processing).

Over the years, Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers and professionals—have gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board was changed to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

What language is Arduino? Arduino code is written in C++ with an addition of special methods and functions, which we'll mention later on. C++ is a human-readable programming language. When you create a 'sketch' (the name given to Arduino code files), it is processed and compiled to machine language.

Arduino IDE, the Arduino Integrated Development Environment (IDE), is the main text editing program used for Arduino programming. It is where you'll be typing up your code before uploading it to the board you want to program. Arduino code is referred to as sketches.

Arduino code example, as you can see, the IDE, has a minimalist design. There are only five headings on the menu bar, as well as a series of buttons underneath which allow you to verify and upload your sketches. Essentially, the IDE translates and compiles your sketches into a code that Arduino can understand. Once your Arduino code is compiled, it is then uploaded to the board's memory. If there are any errors in the Arduino code, a warning message will flag up prompting the user to make changes. Most new users often experience difficulty with compiling because of Arduino's stringent syntax requirements. If you make any mistakes in your punctuation when using Arduino, the code will not be compiled an error message will be displayed.

Figure 6. Arduino code examples

9.1. Serial monitor and serial plotter

Arduino serial monitor can be opened by clicking on the magnifying glass icon on the upper right side of the IDE or under tools. The serial monitor is mainly used for interacting with the Arduino board using the computer, and is a great tool for real-time monitoring and debugging. In order to use the monitor, you'll need to use the Serial class.

The code you download from circuito.io has a test section that helps you test each component using the serial monitor, as you can see in the following screenshot:

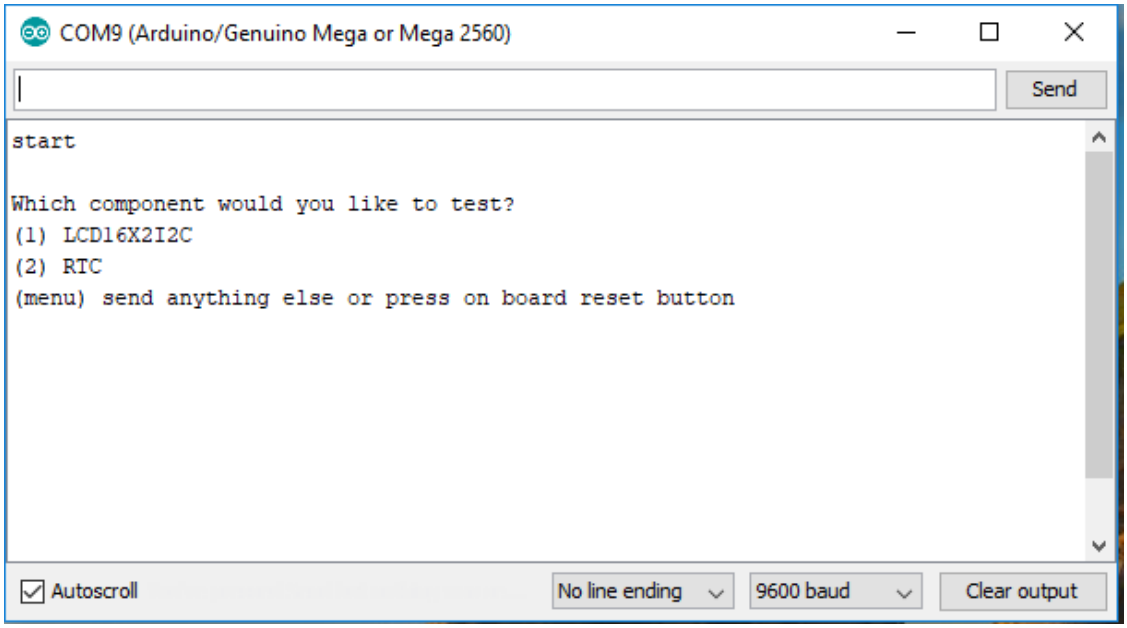

Figure 7. Test section

Arduino serial plotter is another component of the Arduino IDE which allows to generate a real-time graph of the serial data and negative value graphs, and conduct waveform analysis. The serial plotter makes it much easier to analyse the data through a visual display.

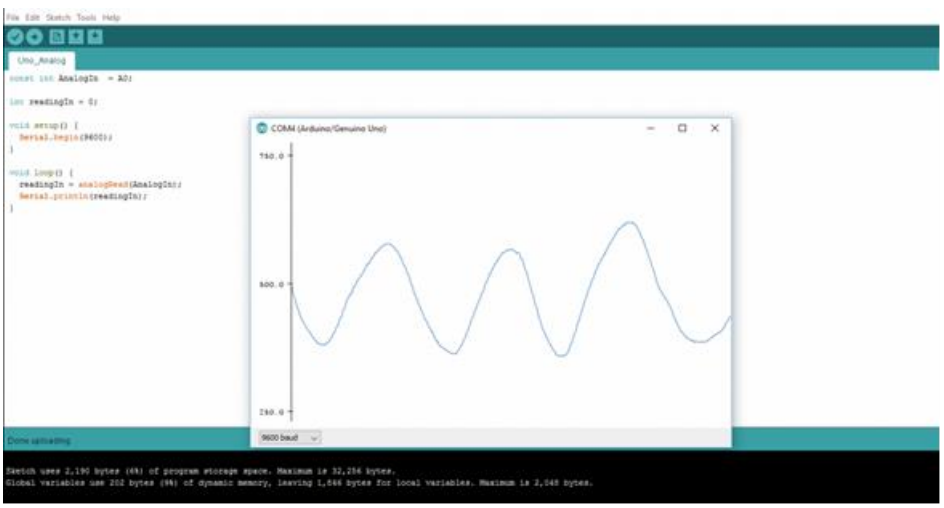

Figure 8. Serial plotter

9.2. Debugging Arduino code and hardware

Unlike other software programming platforms, Arduino doesn't have an on-board debugger. Users can either use a third-party software or utilize the serial monitor to print Arduino's active processes for monitoring and debugging. By using the serial class, debugging comments and values of variables are displayed on the serial monitor. On most Arduino models, this will be done using serial pins 0 and 1 which are connected to the USB port.

9.3. Code structure

9.3.1. Libraries

In Arduino, much like other leading programming platforms, there are built-in libraries that provide basic functionality. In addition, it's possible to import other libraries and expand the Arduino board capabilities and features. These libraries are roughly divided into libraries that interact with a specific component or those that implement new functions.

To import a new library, you need to go to Sketch > Import Library.

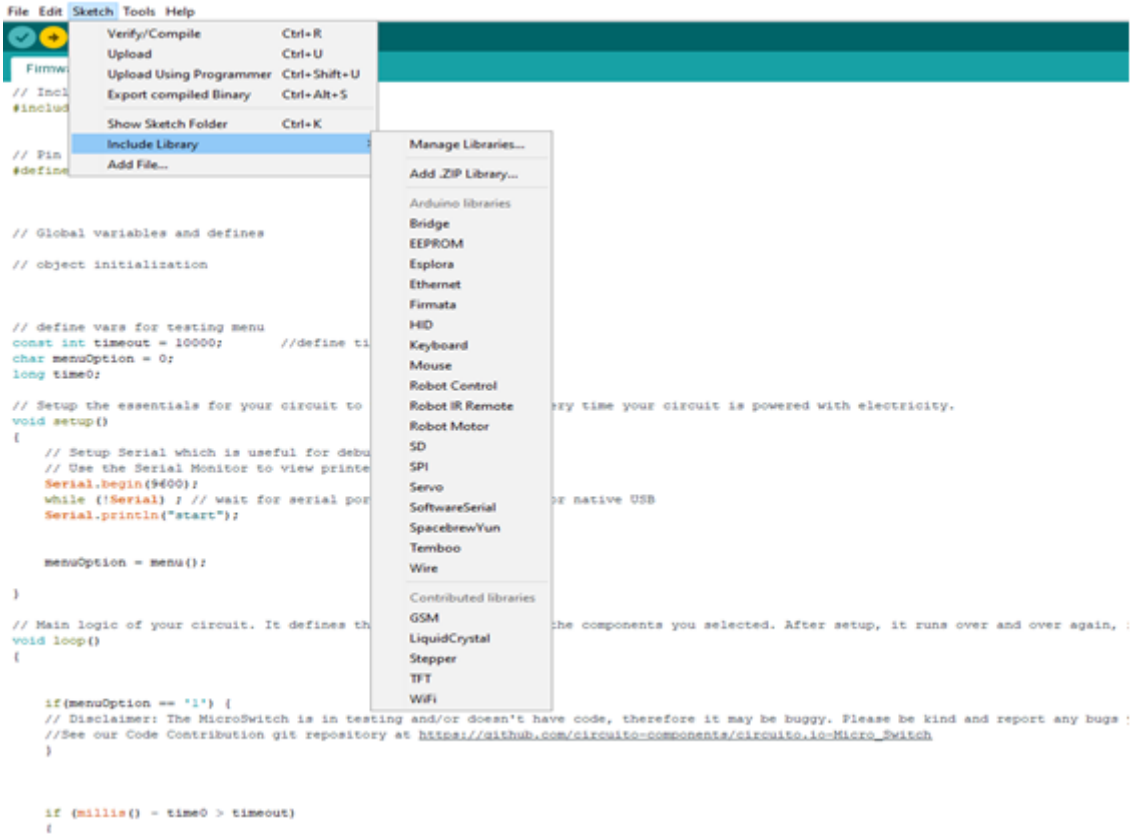

Figure 9. Libraries of Arduino

In addition, at the top of your .ino file, you need to use '#include' to include external libraries. You can also create custom libraries to use in isolated sketches. Pin Definitions: to use the Arduino pins, you need to define which pin is being used and its functionality. A convenient way to define the used pins is by using: '#define pin Name pin Number'. The functionality is either input or output and is defined by using the pin Mode () method in the setup section.

9.3.2. Declarations—variables

While using Arduino, global variables and instances to be used later on have to be declared. In a nutshell, a variable allows to name and store a value to be used in the future. For example, data acquired from a sensor can be stored in order to use it later. To declare a variable, simply define its type, name and initial value.

It's worth mentioning that declaring global variables isn't an absolute necessity. However, it's advisable that you declare your variables to make it easy to utilize your values further down the line.

9.3.3. Instances

In software programming, a class is a collection of functions and variables that are kept together in one place. Each class has a special function known as a constructor, which is used to create an instance of the class. In order to use the functions of the class, we need to declare an instance for it.

Setup ()

Every Arduino sketch must have a setup function. This function defines the initial state of the Arduino upon boot and runs only once.

Here, we'll define the following:

1. Pin functionality using the pin Mode function
2. Initial state of pins
3. Initialize classes
4. Initialize variables
5. Code logic

Loop ()

The loop function is also a must for every Arduino sketch and executes once setup () is complete. It is the main function and, as its name hints, it runs in a loop over and over again. The loop describes the main logic of the circuit.

For example:

Figure 10. Loops

Note: The use of the term 'void' means that the function doesn't return any values.

9.3.4. How to program Arduino

The basic Arduino code logic is an 'if-then' structure and can be divided into four blocks:

Setup—will usually be written in the setup section of the Arduino code, and performs things that need to be done only once, such as sensor calibration.

Input—at the beginning of the loop, reads the inputs. These values will be used as conditions ('if') such as the ambient light reading from an LDR using analogRead().

Manipulate Data—this section is used to transform the data into a more convenient form or perform calculations. For instance, the AnalogRead() gives a reading of 0–1,023 which can be mapped to a range of 0–255 to be used for PWM.(see analogWrite())

Output—this section defines the final outcome of the logic ('then') according to the data calculated in the previous step. Looking at our example of the LDR and PWM, turn on an LED only when the ambient light level goes below a certain threshold.

9.4. Arduino Code libraries

9.4.1. Library structure

A library is a folder comprised of files with C++ (.cpp) code files and C++ (.h) header files.

The .h file describes the structure of the library and declares all its variables and functions.

The .cpp file holds the function implementation.

9.4.2. Importing libraries

The first thing you need to do is find the library you want to use out of the many libraries available online. After downloading it to your computer, you just need to open Arduino IDE and click on Sketch > Include Library > Manage Libraries. You can then select the library that you want to import into the IDE. Once the process is complete, the library will be available in the sketch menu.

In the code provided by circuito.io instead of adding external libraries like mentioned earlier, we provide them with the firmware folder. In this case, the IDE knows how to find them when using #include.

9.4.3. From Software to Hardware

There is a lot to be said of Arduino's software capabilities, but it's important to remember that the platform is comprised of both software and hardware. The two work in tandem to run a complex operating system.

Code → Compile → Upload → Run

At the core of Arduino is the ability to compile and run the code.

After writing the code in the IDE, upload it to the Arduino. Clicking the Upload buttons (the right-facing arrow icon) will compile the code and upload it if it passed the compilation. Once the upload is complete, the program will start running automatically.

You can also do this step by step:

1. First, compile the code. To do this, simply click the check icon (or click on sketch > Verify / Compile) in the menu bar.

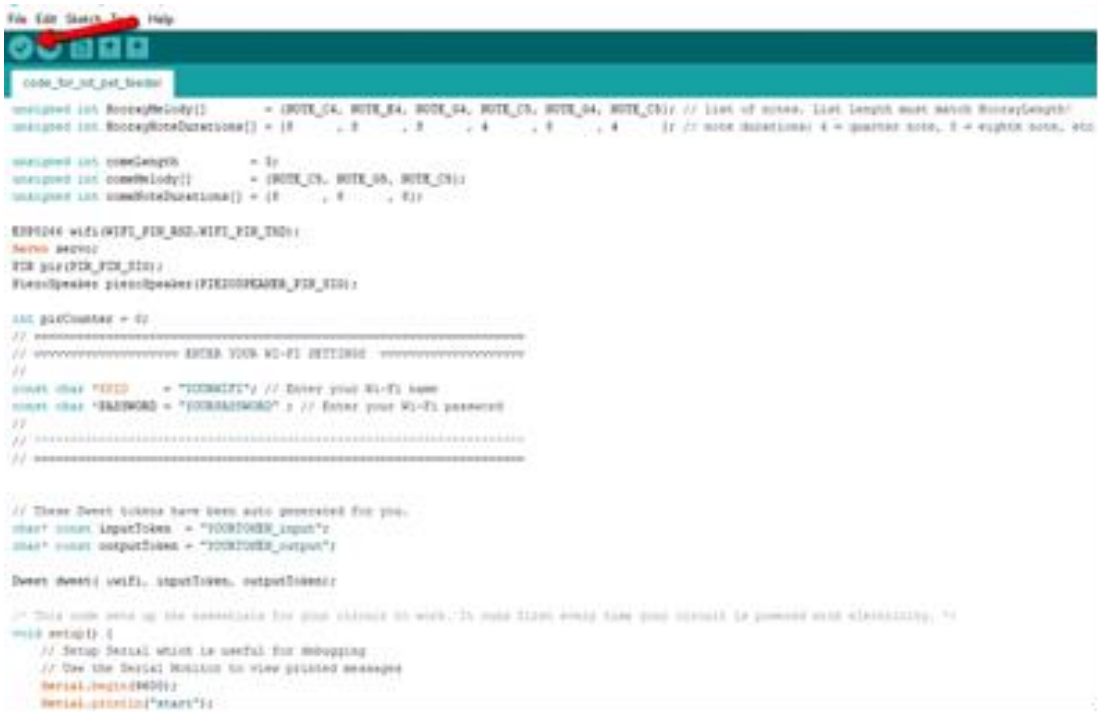

Figure 11. Checker

As you can see, the check icon is located in the top left underneath the 'File' tag in the menu section.

Once you've done this, Arduino will start to compile. Once it's finished, you'll receive a completion message that looks like this:

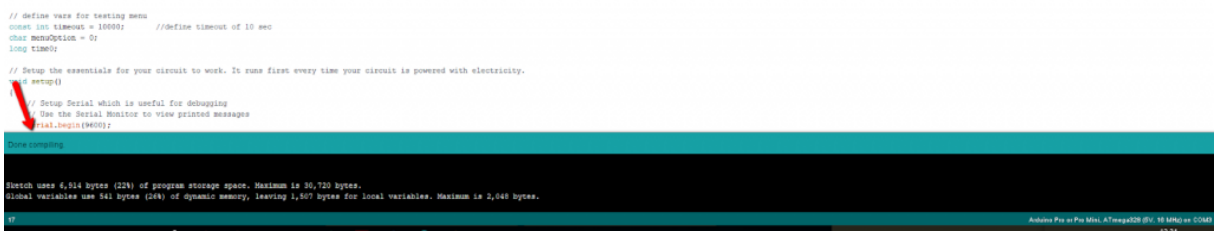

Figure 12. Compiling

As you can see, the green line at the bottom of the page tells you that you're 'done compiling'. If your code fails to run, you'll be notified in the same section, and the problematic code will be highlighted for editing.

Once you've compiled your sketch, it's time to upload it.

1. Choose the serial port your Arduino is currently connected to. To do this, click on Tools > Serial port in the menu to designate your chosen serial port (as shown earlier). You can then upload the compiled sketch.
2. To upload the sketch, click on the upload icon next to the tick. Alternatively, you can go to the menu and click File> upload. Your Arduino LEDs will flicker once the data are transferred.

Once complete, you'll be greeted with a completion message that tells you that Arduino has finished uploading.

9.4.4. Setting up your IDE

In order to connect an Arduino board to your computer, you need a USB cable. When using the Arduino Uno, the USB transfers the data in the program directly to your board. The USB cable is used to power your Arduino. You can also run your Arduino through an external power source.

Before uploading the code, there are some settings that need to be configured.

Choose your board - Designate the Arduino board that is ready to use. Do this by clicking Tools > Board > Your Board.

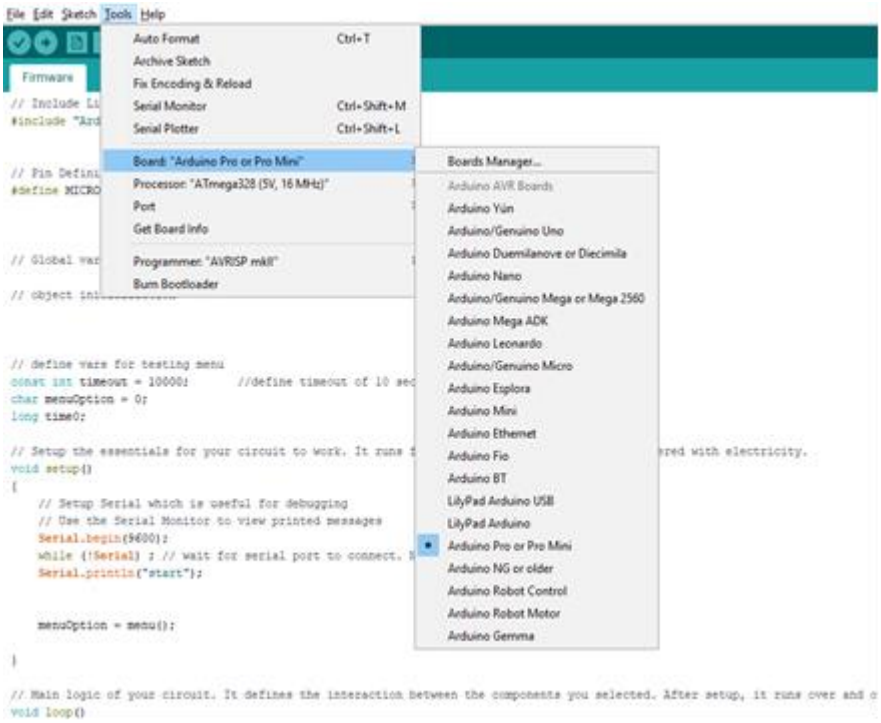

Figure 13. Choosing the right board

Choose your processor- there are certain boards (for example Arduino pro-mini) for which you need to specify which processor model you have. Under tools > processor > select the model you have.

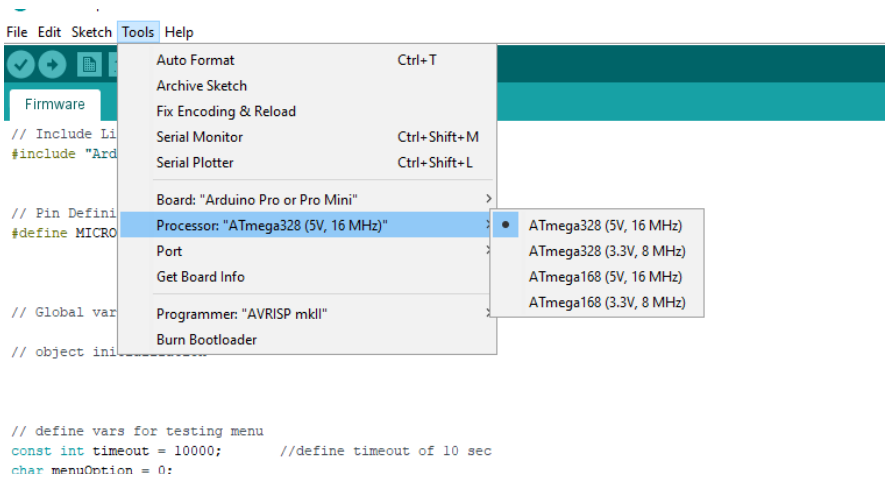

Figure 14. Choosing your processor

Choose your port—to select the port to which your board is connected, go to tools > Port > COMX Arduino (This is Arduino's serial port).

How to Install Non-Native Boards (e.g., NodeMCU)

Some board models are not pre-installed in the Arduino IDE; therefore, you'll need to install them before you can upload the code.

To install a non-native board such as NodeMCU, you need to:

1. Click on tools > Boards > Boards Manager
2. Search for the board you want to add in the search bar and click 'install'.

Some boards cannot be found through the Board Manager. In this case, you'll need to add them manually. In order to do this:

1. Click on Files > Preferences
2. In the Additional Boards Manager field, paste the URL of the installation package of your board. For instance, for nodeMCU, add the following URL:

   http://arduino.esp8266.com/stable/package_esp8266com_index.json
3. Click OK
4. Go to tools > Boards > Boards Manager
5. Search for the board you want to add in the search bar and click 'install'.

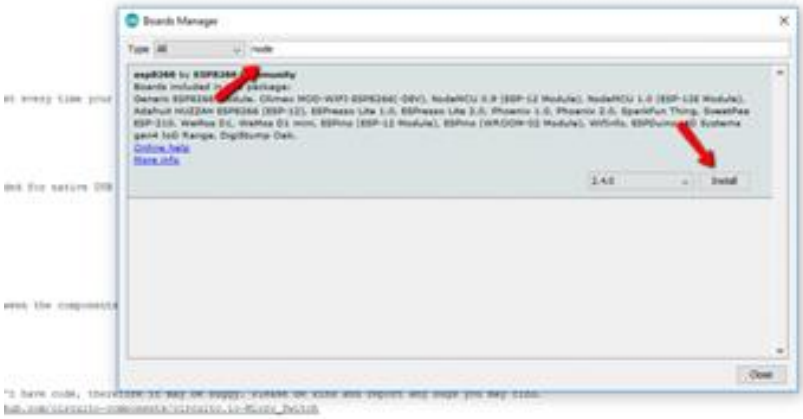

Figure 15. Board manager

Once you've completed this step, you will see the installed boards in the boards' list under tools.

Note: the process may differ slightly for different boards.

Arduino: An Extremely Versatile Platform

Arduino is far more than just a simple microcontroller. With an expansive IDE and a vast array of hardware configurations, Arduino is truly a diverse platform. The variety of its libraries and its intuitive design makes it a favourite for new users and experienced makers alike. There are thousands of community resources to help you get started with both hardware and software.

As you advance your skills, you may face problems that require debugging, which is a weak spot of the Arduino IDE. Luckily, there are several tools and methods to debug Arduino hardware and software. In the next article, we're going to look at how to debug Arduino (and how to test Arduino code) as well as how to use simulators and emulators.

10. Result and discussions

The intelligent walking stick, built with the highest degree of precision, will assist the blind individuals move from one location to another without helping others. This could also be seen as a crude manner to give a feeling of vision to the blind. This stick decreases the reliance on other family members, friends, and guide dogs of visually impaired people while wandering around. The suggested combination of different work units allows a system that tracks the user's position in real time and offers dual feedback that makes navigation safer and more secure. The intelligent stick detects objects or barriers in front of users and feeds warning back, rather than vibration in the form of speech messages.

11. Conclusion

Finally, the Blind Walking Stick was produced into a prototype that can be used to guide the blind. Its objective is to fix the issues experienced in their daily lives by the blind individuals. To guarantee their safety, the system also requires the measure. This project will work to assist all the blind individuals around the world to make it simple for them to walk wherever they want and move forward. It is used to assist blind individuals with disabilities to promote motion and enhance safety.

References

Balakrishnan, G., Sainarayanan, G., Nagarajan, R., & Yaacob, S. (2007). Wearable real-time stereo vision for the visually impaired. *Engineering Letters, 14*(2).

Dodds, A., Clark-Carter, D. & Howarth, C. (1984). The sonic PathFinder: an evaluation. *Journal of Visual Impairment and Blindness, 78*(5), 206–207.

Grum, D. K. & Grum, B. (2017). Visual art education for blind persons: psychological perspective. *New Trends and Issues Proceedings on Humanities and Social Sciences, 2*(4). Retrieved from https://un-pub.eu/ojs/index.php/pntsbs/article/view/1147

Kuranov, A. (2002). An empirical analysis of boosting algorithms for rapid objects with an extended set of haar-like features. Intel Tech. Rep.

Lienhart, R. & Maydt, J. (2002, September). *An extended set of haar-like features for rapid object detection* (vol. 1, pp. I–I). Proceedings. international conference on image processing. IEEE.

Mostafa, A. F. & Mta'en, I. B. Y. B. (2017). Effect of two different designs of screen readers' programs on developing using the Internet skills of blind middle school students. *World Journal on Educational Technology: Current Issues, 9*(2), 98–111. https://doi.org/10.18844/wjet.v9i2.545

Sathya, D. & Kumar, P. G. (2017). Secured remote health monitoring system. *Healthcare Technology Letters, 4*(6), 228–232.