

Design and analysis of automatic basketball score counter using arduino in Rwanda.

Author: BIMENYIMANA Emmanuel

E-mail: bimenyiem@gmail.com

ABSTRACT

The automatic basketball score counter is the system implemented to count the basketball goals when a team score a goal. This is achieved by using push buttons, Arduino and 7 segments display. When a team score a goal and referee accept it by pushing on up push button, Arduino send the command to the 7 segment and it counts one goal. As you continue to push that push button the 7 segment will add one goal to the goals of scored team .When scored goal create confusion and referee decide to refuse it by pushing on down push button, the 7 segment will remove one goal to the goals of a team. This study carried out in Kigali city of Rwanda intended to the design of the automatic basketball score counter machine using arduino in order to prevent the conflict between the two team and to reduce the time. The objective was to design the basketball score counter which will operate automatically. The findings have shown that the best alternative is to use the arduino as a command. The project work concludes that if the automatic basketball score counter was used in national basketball competition and help for reducing misunderstanding about goal counting.

Keywords: automatic counter, arduino, push button, segment

1. BACKGROUND

Basketball was originally played with a soccer ball. These round balls from "association football" were made, at the time, with a set of laces to close off the hole needed for inserting the inflatable bladder after the other sewn-together segments of the ball's cover had been flipped outside-in. These laces could cause bounce passes and dribbling to be unpredictable. Eventually a lace-free ball construction method was invented, and this change to the game was endorsed by Naismith. (Whereas in American football, the lace construction proved to be advantageous for gripping and remains to this day.) The first balls made specifically for basketball were brown, and it was only in the late 1950s that Tony Hinkle, searching for a ball that would be more visible to players and spectators alike, introduced the orange ball that is now in common use. Dribbling was not part of the original game except for the "bounce pass" to teammates. Passing the ball was the primary means of ball movement. Dribbling was eventually introduced but limited by the asymmetric shape of early balls. Dribbling was common by 1896, with a rule against the double dribble by 1898.

The match in sports is held for the specified time of the sporting event, for example the length of time in a soccer match is for 90 minutes. The timers used are often used by watches or stopwatches that can only be seen by the referee and some are using digital clocks displayed on a large display. In addition to the

length of time specified in a sporting event, also recorded the score obtained by each team. These scores are recorded manually or digitally displayed on large displays to be viewable by all spectators of matches around the match field. Usually each type of sport requires a separate scoreboard for score calculation and there is also a scoreboard while displaying the game time. For cost efficiency, it is designed and made scoreboard that can be used for some kind of sports match based on microcontroller which can be controlled by using cable or also can be controlled from certain distance without cable by using Bluetooth. This scoreboard can be used for several sports games such as basketball, futsal, ping pong, badminton, and others. This scoreboard uses a power supply that is backed up battery power that can be automatically charged. This scoreboard can also be easily moved. This scoreboard has a high accuracy because both timing and score calculation is done by using a microcontroller, this scoreboard has been used by student activity units in the Universitas HKBP Nommensen Medan, Indonesia.

2. METHODS

This research applies mixed methods, which relates to the analysis of social, economic and spatial phenomena. This study analyses only secondary data. In addition, it uses both quantitative and qualitative data gathered from secondary sources including textbooks, academic journals, working

papers, organisations reports, policy documents and laws and other common forms of publications.

$$R=R1=R2=R3=R4=R5=R6=10K$$

Resistor for led protection

The resistor is there to limit the LED's current.

3. RESULTS

In this circuit we will use the resistor for two kinds of resistors

Resistor for led protection of overcurrent and the pull-up resistor.

Pull up resistor: Pull-up resistors are used in electronic logic circuits to ensure that inputs to the Arduino settle at expected logic levels if external devices are disconnected or high impedance. 'Just because you have nothing at all connected to an input pin doesn't mean it is a logical zero.

Our circuit can be supplied by a voltage in the range of 5-12v and here we use a supply of 5 volts and a current of 2A. A certain value of current needs to be limited so that the input pin of Arduino receives a required value, and to limit this current a pull up resistor is introduced, and its value depends on the current needs to be limited. Pull up resistor also prevents a short circuit when the button is pressed.

Supply voltage: 5v

Supply current: 2A

Needed current to be limited: 0.5mA

Resistor value will be obtained using ohm's law

$R=V/I$ where R: resistance, V: voltage, and I: current

Then, $R=5/0.5 \times 10^{-3} = 10000\Omega$

$$V=I \cdot R$$

$$R=V/I$$

Arduino runs at 5V. But not all that will go over the resistor. The LED also has a voltage drop, typically around 2V for a red LED. So there remains 3V for the resistor. A typical indicator LED will have a nominal current of 20mA. In our circuit two LEDs are in parallel then,

$$R = \frac{5V - 2V}{0.02} = 150\Omega$$

4.1.1.5: SEVEN SEGMENT DISPLAY

If your Arduino application only needs to display numbers, consider using a seven-segment display. The seven-segment display has seven LEDs arranged in the shape of number eight. They are easy to use and cost effective. In our circuit we use common anode seven segment display, on these seven segments display the positive pin are connected together and connected to the vcc of 5v.

4.1.1.6: OVERALL BLOCK DIAGRAM

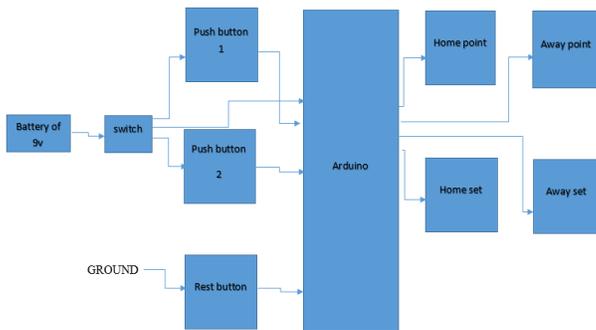


figure 1: Block diagram of project

4. DISCUSSION

Arduino Software

Software programs, **called sketches**, are created on a computer using the Arduino **Integrated Development Environment (IDE)**. The IDE enables you to write and edit code and convert this code into instructions that Arduino hardware understands. The IDE also transfers those instructions to the Arduino board (a process called uploading).

Source code containing computer instructions for controlling Arduino functionality is usually referred to as a **sketch** in the Arduino community. **The word sketch will be used throughout this book to refer to Arduino program code.**

The IDE is split up into the Toolbar across the top, the code or Sketch Window in the Centre and the Serial Output window at the bottom.

The Toolbar consists of 7 buttons; underneath the Toolbar is a tab, or set of tabs, with the filename of

the code within the tab. There is also one further button on the far right hand side. Along the top is the file menu with drop down menus headed under File, Edit, Sketch, Tools and Help. The buttons in the Toolbar provide convenient access to the most commonly used functions within this file menu.



The **Upload** to I/O Board button will upload the code within the current sketch window to your Arduino. You need to make sure that you have the correct board and port selected (in the Tools menu) before uploading. It is essential that you Save your sketch before you upload it to your board in case a strange error causes your system to hang or the IDE to crash. It is also advisable to Verify/Compile the code before you upload to ensure there are no errors that need to be debugged first.

The **New** button will create a completely new and blank Sketch read for you to enter code into. The IDE will ask you to enter a name and a location for your Sketch (try to use the default location if possible) and will then give you a blank Sketch ready to be coded. The tab at the top of the Sketch will now contain the name you have given to your new sketch.

The **Open** button will present you with a list of Sketches stored within your sketchbook as well as a list of Example sketches you can try out with various peripherals once connected.

Review University Without Borders for the Open Society (RUFSO)

ISSN: 2313-285X

Volume 21, 2020

The **Save** button will save the code within the sketch window to your sketch file. Once complete you will get a done Saving message at the bottom of the code window.

The **Verify/Compile** button is used to check that your code is correct, before you upload it to your Arduino.

The **Stop** button will stop the Serial Monitor from operating. It will also un-highlight other selected buttons. Whilst the Serial Monitor is operating you may wish to press the Stop button to obtain a snapshot of the serial data so far to examine it. This is particularly useful if you are sending data out to the Serial Monitor quicker than you can read it.

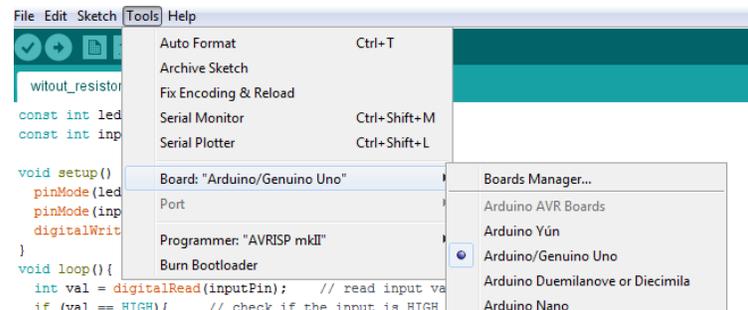
The Serial Monitor is a very useful tool, especially for debugging your code. The monitor displays serial data being sent out from your Arduino (USB or Serial board). You can also send serial data back to the Arduino using the Serial Monitor. If you click the Serial Monitor button, you will be presented with an image like the one above.

On the left hand side, you can select the Baud Rate that the serial data is to be sent to/from the Arduino. The Baud Rate is the rate, per second, that character (data) is sent to/from the board. The default setting is 9600 bauds, which means that if you were to send a text novel over the serial communications line (in this case your USB cable) then 9600 letters, or symbols, of the novel, would be sent per second.

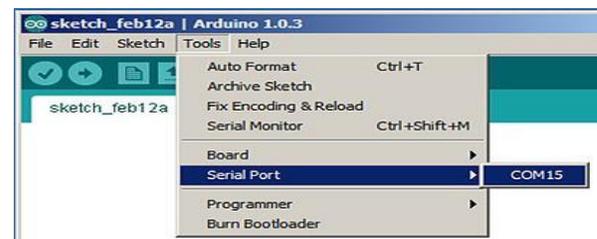
You'll use this IDE (Integrated Development Environment) to make Software VISIBLE! With it you will develop your own software to make Arduino into what you want it to be.

FIRST, YOU MUST SET THE CORRECT BOARD TYPE AND SERIAL PORT

BOARD: In the Arduino IDE top menu click <Tools> and then <Board>. You will see a drop-down menu that looks like this (left). Selection of board for the Arduino Uno click Arduino Uno



SERIAL PORT: Now click <Tools> and then <Serial Port>. On the PC There will usually be only one choice. Selection of Serial Port.



Arduino Hardware

Review University Without Borders for the Open Society (RUFSO)

ISSN: 2313-285X

Volume 21, 2020

The Arduino board is where the code you write is executed. The board can only control and respond to electricity, so specific components are attached to it to enable it to interact with the real world. These components can be sensors, which convert some aspect of the physical world to electricity so that the board can sense it, or actuators, which get electricity from the board and convert it into something that changes the world.

Examples of sensors include switches, accelerometers, and ultrasound distance sensors. Actuators are things like lights and LEDs, speakers, motors, and displays.

There are a variety of official boards that you can use with Arduino software and a wide range of Arduino-compatible boards produced by members of the community (roseMary, 2018). Some of them are given below:

Arduino Mega Board

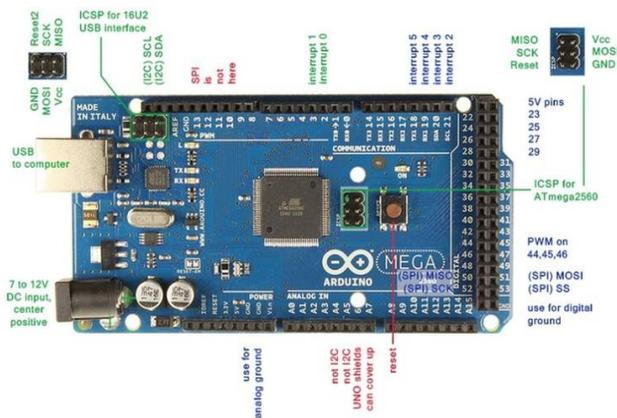


Figure 2: Arduino Mega parts
(<https://www.adafruit.com/product/191>)

will use the power provided by the USB cable when the cable is connected to a computer. If necessary, it can also be powered by an external power supply such as a battery. When you upload code from your computer, the program is saved to the microcontroller itself. This means you can disconnect the Arduino and allow it to function as an independent device. (Smith, 2011), (smith, 2006)

5. CONCLUSION

In conclusion, from the results obtained above, the implemented project of Automatic Basketball score counter using Arduino, comes as solution with more benefits for the use of counting the goals by using digital system without the need of writing by hands. It is flexible to use and as you want to use that system of Basketball score counter in your playground and also it is cheaper for everyone who need it and this project will be competent on the market.

6. LIST OF ABBREVIATIONS

FERWABA	Rwanda Basketball Federation
IDE	Integrated Development Environment
SSD	A seven-segment display
LED	Light Emitting Diode

The Arduino Mega is a good choice for your first Arduino. It is inexpensive and reliable. The Mega

Review University Without Borders for the Open Society (RUFSSO)

ISSN: 2313-285X

Volume 21, 2020

7. ACKNOWLEDGEMENT

This work would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I would like to express my deep gratitude to everyone who have willingly helped me out with their abilities.

Authors' contributions

JR conceptualize the idea and both others contributed equally thereafter.

Authors' affiliation

Author is affiliated to Distance production House University/ IST Burkina Faso.

Conflict of interest

We declare not conflict of interest

Funding

None

REFERENCES

- 1) IC Voltage regulator. (2003). Retrieved from www.rakeshmondal.com: www.rakeshmondal.info/I7805-Voltage-Regulator
- 2) Resistor. (2008). Retrieved from www.learnsparklum.com: <http://sfe.io/t64>
- 3) Resistor. (2008). Retrieved from www.token.com: www.token.com.tw/pdf/resistor/what-is-a-resistor.pdf
- 4) Push-button. (2014). Retrieved from www.futureelectronics.com: <http://www.futureelectronics.com/en/switches/push-button-switches.aspx>
- 5) Fundamentals of seven segment display. (2023, 10). Retrieved from <https://www.electronics-tutorials.ws/wp-content/uploads/2013/10/segment4.gifs>
- 6) Bill, C. (2013). Fundamentals of automation technology//PUSH BOTTON.
- 7) Boylestad, R. (2014). electronic device. In R. Boylestad, electronic device. upper saddle river, new jersey columbus: prentice hall.
- 8) Fixed Voltage reguator. (n.d.). Retrieved from www.circuitstoday.com: <http://www.circuitstoday.com/ic-voltage-regulators#switching-regulator>
- 9) KAISER, C. J. (2011). resistor hand book//variable resistor. (s. edition, Ed.)
- 10) Neumann, N. (2010, 11 19). fundamental of seven segment led display.
- 11) Pumplin, j. (2008). current and resistance.
- 12) RoseMary. (2018, april 6). 8051 microcontroller. Retrieved from www.engineersgarage.com: <https://www.engineersgarage.com/8051-microcontroller>
- 13) Smith, A. g. (2011). arduino project hand book//Introduction to Arduino.
- 14) smith, M. g. (2006). Arduino Main Hardware.
- 15) Wood, V. (19 Nov. 2010). Numitron Readout: Simplified Seven-Segment Display in One Tube. (e. 1, Ed.)