International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319-6734, ISSN (Print): 2319-6726
www.ijesi.org ||Volume 9 Issue 5 Series II || May 2020 || PP 34-39

An Architecture of a Touchscreen Operated Window Blinds for Smart Home

Kenneth Oliver, Michael Herter, and Tareq Khan
School of Engineering, Eastern Michigan University, Michigan, Ypsilanti, 48197, United States

ABSTRACT:*A smart home refers to a convenient home setup where appliances and devices can be automatically controlled remotely using a mobile or other networked device.In this paper, the design of smart blinds is proposed - that will enable a house to become smarter and more energy-efficient. The proposed smart blinds are opened or closed with the push of a button on a wall-mounted touchscreen. The touchscreen is interfaced with a Raspberry Pi and it communicates with each Arduino devices – that are interfaced with each blind of the house - using Bluetooth Low Energy (BLE). A prototype of the touchscreen operated smart blind system is developed and tested successfully.*

KEYWORDS –*Arduino, Blinds, Bluetooth Low Energy (BLE), Graphical User Interface (GUI), Raspberry Pi, Smart Home, Stepper Motor.*

Date of Submission: 06-05-2020 Date of Acceptance: 20-05-2020

I. INTRODUCTION

A smart home allows homeowners to control appliances, thermostats, lights, and other devices remotely using a smartphone or other connected devices.Smart homes can implement either wireless or wired systems—or both. Wireless systems are easier to install, but installing wired systems is more difficult. The global home automation market was valued at about $24 billion in 2016. It is expected to grow to about $53.5 billion by 2022as more people begin to adopt smart home technology [1]. In this paper, we created a smart blind - that allowsus to open and close the blinds in a home with a touch screen panel. The communication from the touch panel to the blinds is wireless – thus it is easier to install. The reason that we believe this product is important is that it is one more step toward a smarter, more functional home.

II. RELATED WORKS

There exist several products that are similar to our product however most of them lack a few key functions or the price points are very high compared to what we set out to accomplish. IKEA sells a smart blind [2], however, their smart blind is a simple roller blind while we set out to use a cellular blind for improved energy efficiency. According to Blind Chalet, as much as 50% of heat energy is lost through windows and cellular shades have one of the highest R-values, a rating of the resistance of heat loss, of any blind design[3]. With these smart blinds being easier to open and close it can lead to energy savings. Another popular company for smart blinds is Lutron [4]. They do provide a small 24” × 24” cellular smart blind, however, it costs over $400. We set out to design our cellular shade with an individual blind cost around $60, a more approachable price point. There have been several attempts at Do It Yourself (DIY) smart blinds [5], however, most of them use a simple horizontal blind and focus on rotating the slats rather than retracting the blind. Compared with other works - our proposed work has a graphical user interface (GUI) on a wall-mounted touchscreen to control the blinds, has wireless communication which makes it easier to install, can retract the blind and it is low cost.

III. SYSTEM ARCHITECTURE

The architecture of the system is shown in Fig. 1. The main control of the system comes from a Raspberry Pi touch screen. The Raspberry Pi communicates via Bluetooth to the Arduino Nanos. Then each of the Arduino Nanos directly controls the stepper motors that are built into the blinds. Each blind has its own Arduino Nano to provide wireless control for the blinds.

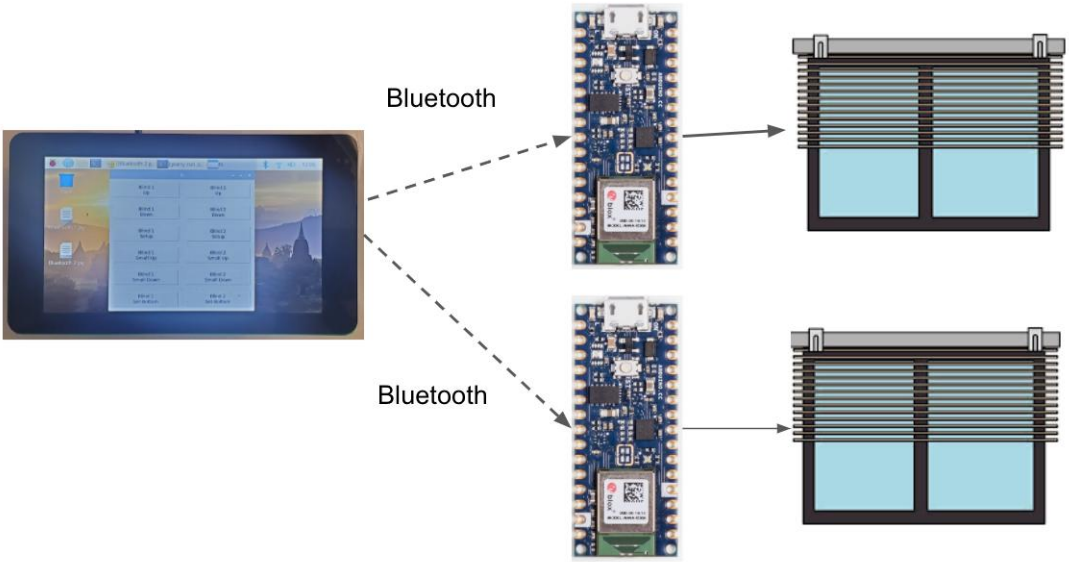


Figure 1: Raspberry Pi touchscreen control center communicates via Bluetooth to an Arduino Nano 33 BLE that in turn controls the stepper attached to the blind.

3.1 Hardware

We decided to make a frame using wood and we cut them to size to fit the blinds inside as shown in Fig. 2.We picked the blinds [6] that are the low cost and can be simply operated. They go up and down, there is no dimming adjustment on the blinds which made it simpler for us in the design of the motor bracket. The motor bracket was designed in *Fusion 360 by AutoDesk*. The drawing in Fig. 3 is the finished product that was then printed with a 3D printer. This piece is then screwed to the frame, in this case, the frame of the blinds. Normally this would be installed inside the wall and not be visible.

Figure 2: The frame for the blinds with blinds installed and brought up to the top.

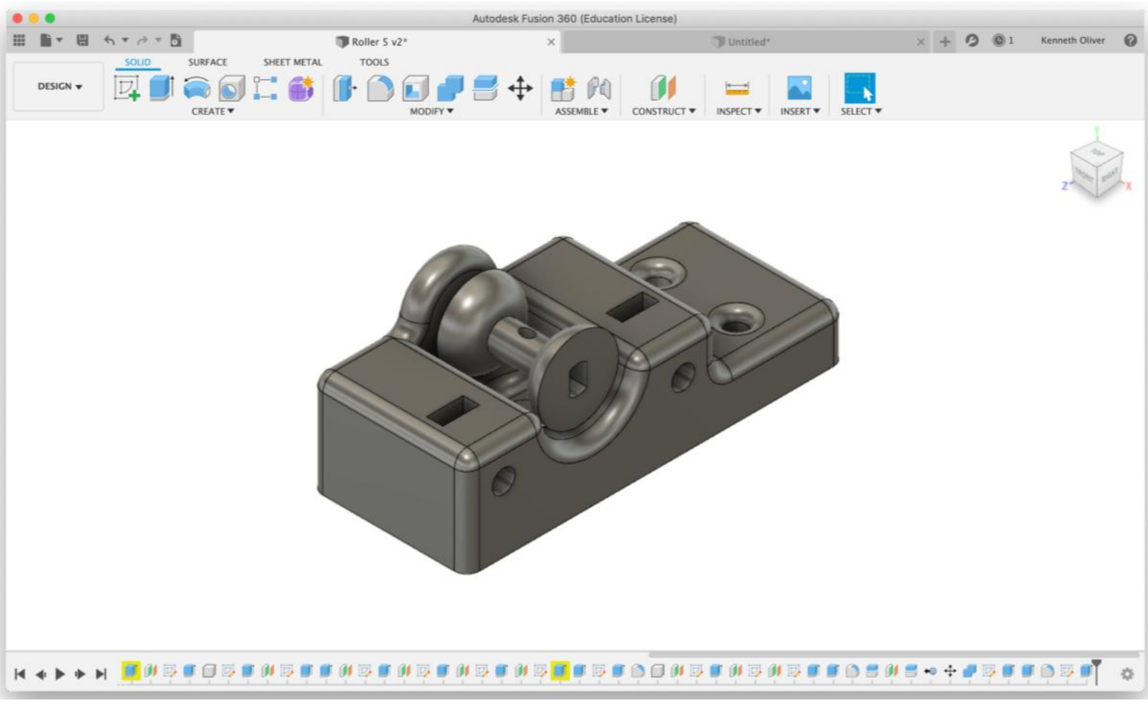


Figure 3: Modeling of the motor bracket for 3D printing.

Fig. 4 shows how the hardware works together. The system starts with a Liquid Crystal Display (LCD) touchscreen [7]. The LCD has a 24-bit color depth and a screen resolution of 800 × 480 pixels. The touchscreen is interfaced with the Raspberry Pi [8] using the DSI and the I2C port. Then the Raspberry Pi talks to the Arduino Nano [9] via a BLE connection that then turns the stepper motor. The Arduino is connected to a stepper motor control board and each control board is connected with a stepper motor. The stepper driver control board is based on the ULN2003 [10].

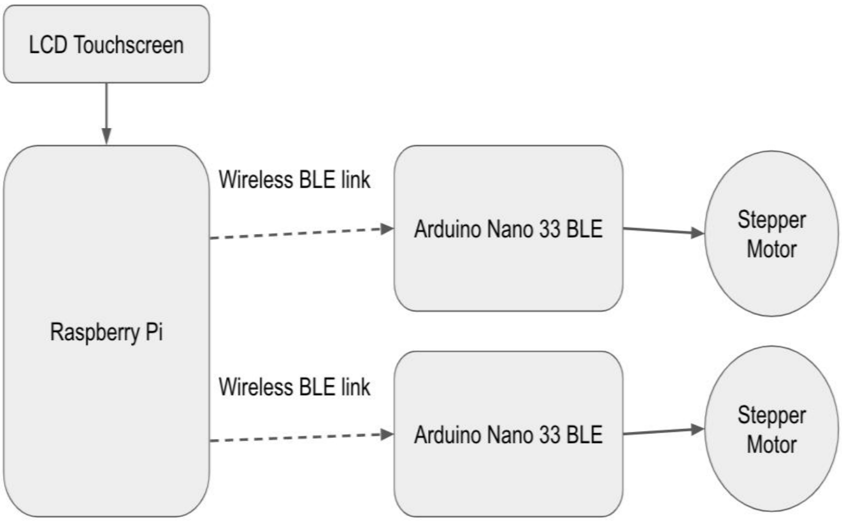


Figure 4: Block Diagram of hardware

3.2 Embedded Firmware

The Raspberry Pi runs a Python script that uses the *Tkinter* library [11] for building the GUI with several buttons. The GUI shows buttons for blind up, blind down, and blind setup. Upon selecting the blind setup, the user is presented with 3 additional buttons for moving the blind to the very bottom of the frame and then defining that as the bottom, this is necessary for the Arduino to track the position of the blind to make the correct movements.

When the user pushes a button on the display, the Raspberry Pi then attempts to connect to the Arduino Nano through a wireless BLE signal. If the connection was successful it sends a value (0 through 4) that corresponds to the desired movement. The Arduino receives the value from the Raspberry Pi and executes a chunk of code that corresponds to the received value.

The Arduino controls the stepper by keeping track of where the blind is at and where the user wants it to go. This makes it necessary to reset the blind to the bottom and define that position after each power cycle of the Arduino Nano. Once the user defines the bottom and assures that the blinds go all the way to the top, there is no issue with them going up and down without recalibration. Fig. 5 shows a flowchart of this firmware.

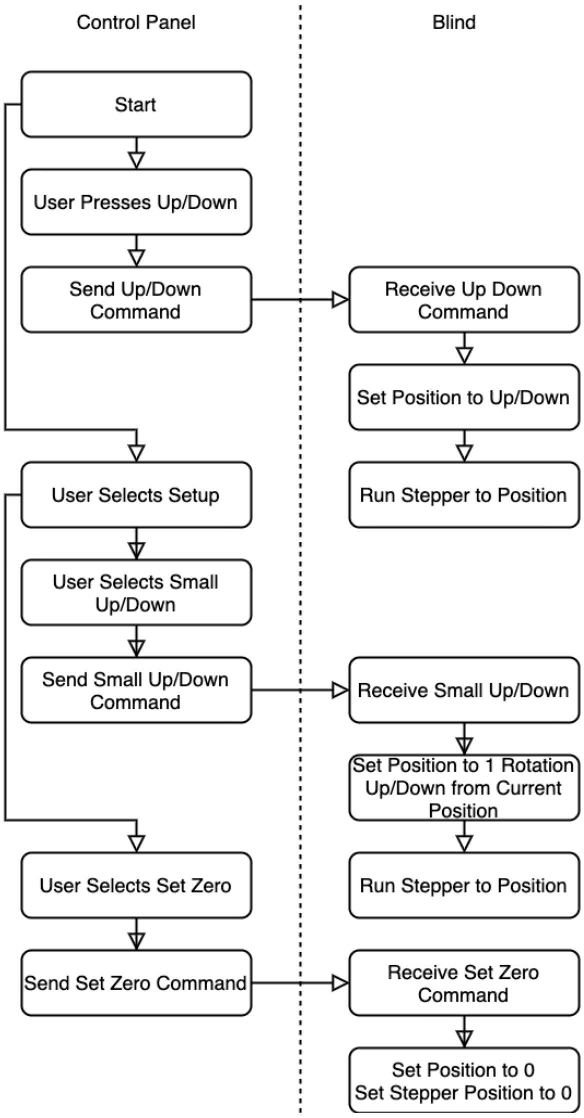
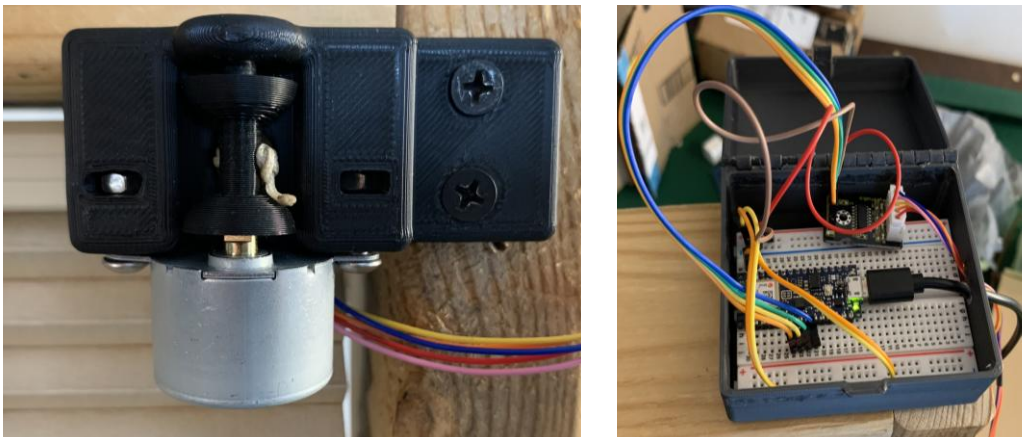


Figure 5: Flow chart of the firmware

IV. RESULTS

A prototype of the proposed system is developed and tested successfully. Fig. 6 shows the overall build of the project with the touch screen control panel in the lower left and the 3D printed motor mount in the top right. Fig. 7(a) shows a close up of the mount and Fig. 7(b) shows the wiring of the Arduino Nano and the stepper control panel.

Figure 6: The smart blind prototype with the touchscreen display.



(a) (b)

Figure 7: (a)Close up of stepper motor mount and spool; (b)**w**iring of Arduino Nano and stepper motor control board placed inside a box.

As seen in Table 1, the largest power consumption comes from the Raspberry Pi with the touch screen. We disabled the Wi-Fi on the Raspberry Pi as we were not using it and it saved a little power however it was still pulling around 1 A current while running the Python script. The Arduino Nano consumes 18 mA current when idle and fluctuates around 50mA while the stepper motor is running. This gives us a total current of 1.018A when idle and 1.05A when one blind is moving. For each additional blind,it will consume another 18mA while in idle, and 50mA while in moving.

Table 1: Current Consumption

Device	**Current (mA)**
Raspberry Pi	1000
Arduino Nano 33 BLE (idle)	18
Arduino Nano 33 BLE (driving motor)	50

V. CONCLUSION

In this paper, a smart blind is designed, developed, and tested. The proposed system uses the state of the art touchscreen – that can be mounted on the wall to control the blinds. It uses wireless communication from the control panel to the blinds – thus installing the system is easier than wired systems. Future work includes using higher-powered 12v stepper motor - this would be beneficial to provide more power for moving larger blinds. We also would like to implementelectrically erasable programmable read-only memory (EEPROM) on the Arduino and then we would be able to store the data that keeps track of where the blind is in the EEPROM. It would allow the Arduino to keep track of the position of the blind through a power cycle. We would also like to implement a Google Home into the control system of the blinds, with the Google Home we could then have a voice control method to the blinds. The Google Home also has routines that the user could enable and would provide the user with the ability to set a time for the blinds to open/close and leads for a further automated process. We would also design a smartphone application in Android that would be able to control the blinds through Bluetooth - allowing the user to control the blinds without having to move to the touchscreen device as most users would carry their phones with them at all times.

ACKNOWLEDGMENTS

We would like to thank Eastern Michigan University for funding this project and for continuing support in difficult times of COVID-19.

REFERENCES

[1]. James Chen, "Smart Home," [Online]. Available: https://www.investopedia.com/terms/s/smart-home.asp, 2020.
[2]. IKEA FYRTUR Blackout Roller Blind - Wireless, Battery Operated Gray, [Online]. Available: www.ikea.com/us/en/p/fyrtur-blackout-roller-blind-wireless-battery-operated-gray-70417463/, 2020.
[3]. "Window Treatments: R-Value and Energy Efficiency." Blinds Chalet, [Online]. Available: www.blindschalet.com/kba-window-treatment-energy-efficient-56.html, 2020.
[4]. "Serena Motorized Shades by Lutron: Smart Device Remote Control." [Online]. Available: www.serenashades.com/, 2020.
[5]. "Smart Blinds," Instructables, [Online]. Available: www.instructables.com/id/Smart-Blinds/, 2017
[6]. Bali Blinds Custom Blackout Cellular Shade with Cord Lift, 3/8" Fabric, Double Cell Midnight Mesa, 20" x 20", [Online]. Available:https://www.amazon.com/dp/B01G5NZCW6, 2020.
[7]. Raspberry Pi 7" Touch Screen Display, [Online]. Available:https://www.amazon.com/dp/B0153R2A9I, 2020.
[8]. CanaKit Raspberry Pi 4 Starter Kit, [Online]. Available:https://www.amazon.com/dp/B07V4G63M1, 2020.
[9]. Arduino Nano 33 BLE, [Online]. Available:https://www.amazon.com/dp/B07WXKWNQF, 2020.
[10]. KEYESTUDIO 3 Set of Stepper Motor + ULN2003 Driver Board, [Online]. Available:https://www.amazon.com/dp/B07DRNQ4PK/, 2020.
[11]. tkinter — Python interface to Tcl/Tk, [Online]. Available:https://docs.python.org/3/library/tkinter.html, 2020.

Kenneth Oliver, et. al. "An Architecture of a Touchscreen Operated Window Blinds for Smart Home." *International Journal of Engineering Science Invention (IJESI),* Vol. 09(05), 2020, PP 34-39.