RE-ENGINEERING OF STRING TRIMMERS' DRIVER & PROTECTION GUARD

Vinesh Thiruchelvam
School of Engineering
Asia Pacific University of Technology & Innovation
Technology Park Malaysia, Bukit Jalil 57000
Kuala Lumpur, Malaysia
Email: dr.vinesh@apu.edu.my

Raffael Fondacci
School of Engineering
Asia Pacific University of Technology & Innovation
Technology Park Malaysia, Bukit Jalil 57000
Kuala Lumpur, Malaysia
Email: raffaelfondacci@gmail.com

ABSTRACT

The existing lawn trimmer model #GL300-XD has some practical down-sides. A durable mounted protection guard and an integrated speed control system with temperature monitoring have been implemented. The design has been effectively realized by using an Arduino as the controller, a solid state relay as the driving tool and an LM35 sensor to monitor the temperature. Coding using the 'analoguewrite' function mapped to the 'digitalwrite' to control the speed of the motor using a potentiometer as the control knob was developed. The sensor reading was incorporated in the coding as a condition to keep system on or off depending on the temperature recorded value. This project was successfully completed, and according to the vibration test, speed test and temperature test done, the results obtained show that both good hardware design and speed response based on temperature reading attained with the implemented system with overall efficiency of 93%.

Keywords: Durable protection guard, LM35 sensing, Arduino control.

Introduction

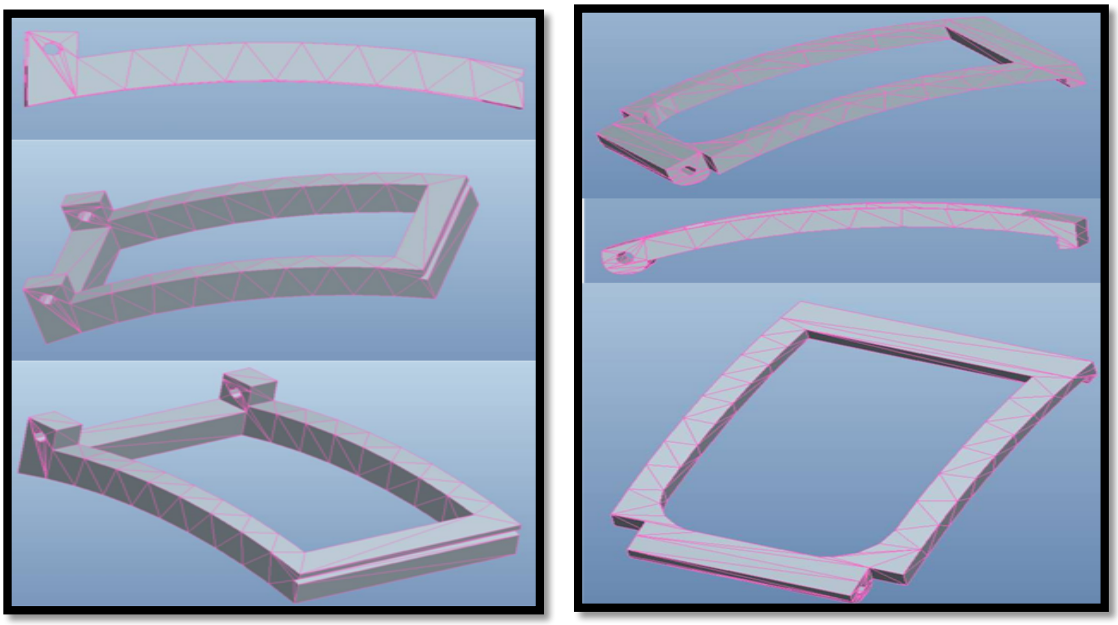
The research was done on an existing operational #GL300-XD string trimmer. This model type works on principles that a line turned fast enough by centrifugal force is held out from its housing and the faster it turns the stiffer the monofilament line. Some monofilament lines designed for more powerful cutters have an extruded shape, like a star. It cuts through faster and more efficiently. Smooth operations of the machine were not taking place with many stoppages based on mechanical or electronics issues.

These types of trimmers are hand held, which means the user has to carry it while operating. This could take from minutes to hours depending on the property size. So one improvement needed to be made was on the weight of the trimmer to make it more comfortable and less exhausting for different users with less time wasting. The second issue was that the motor had only one constant speed and that some terrains need a faster motor run speed in order to cut through the plants, weeds and over grown grass. And in some cases we need a slower motor run speed. This needs improvement, and implementing a speed controller system benefited the controllability of the unit. The final issue in this research is the protection system. The clipped grass was slipping into the motor compartment and creating blockages and burning the motor due to overheating. Designing a new type of guard rectified the issue based on durability and also protection against debris.

SYSTEM DESIGN – MATERIALS AND 3D ENGINEERING

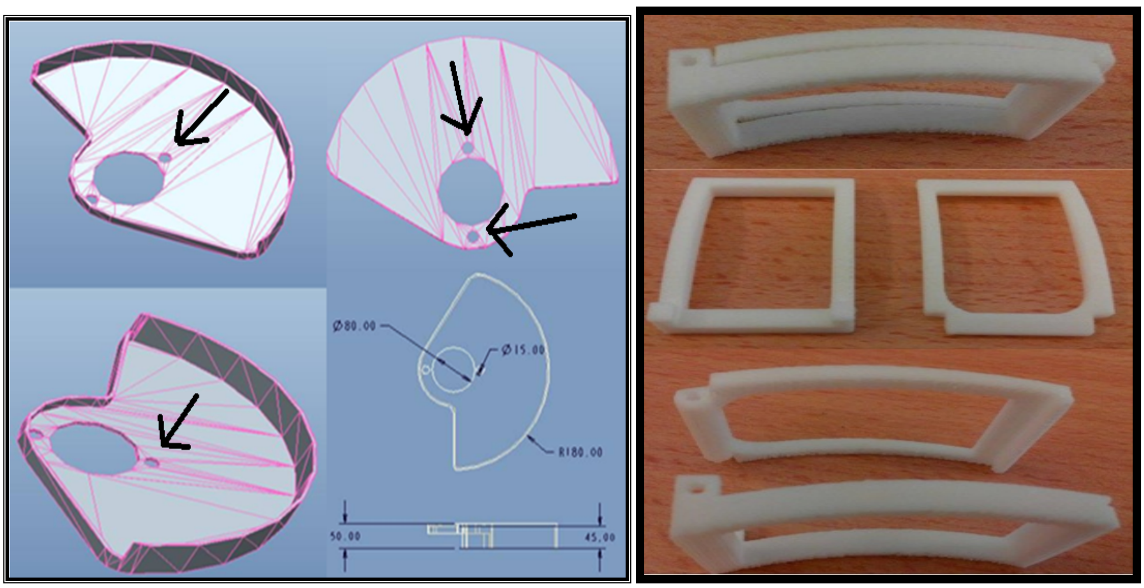
The upper and lower arm netting frame was 3D printed using the abs plastic. It is a framing that will in-case the netting over the motor compartment from the outside. The frame has a lower arm and an upper arm which is coupled at a joint on one end and clipper at the other end. This clip-on and clip-off mechanism makes it easy to clean or replace the netting. The length is 5cm long, 3.4cm of width with 0.4cm thickness. Below is the 3D illustration in different views of the uppr and lower arms.

Figure 1: Lower Arm and Upper Arm Protection Guard 3D Design with Netting Frame



The modification made was analyzed using the Creo software to see if it can overcome the stress and vibrations applied to the protection guard. The 3D illustration and 2D drawing is shown below. The arrows indicate the final modifications made, which is the added feature for mounting strengthening purposes.

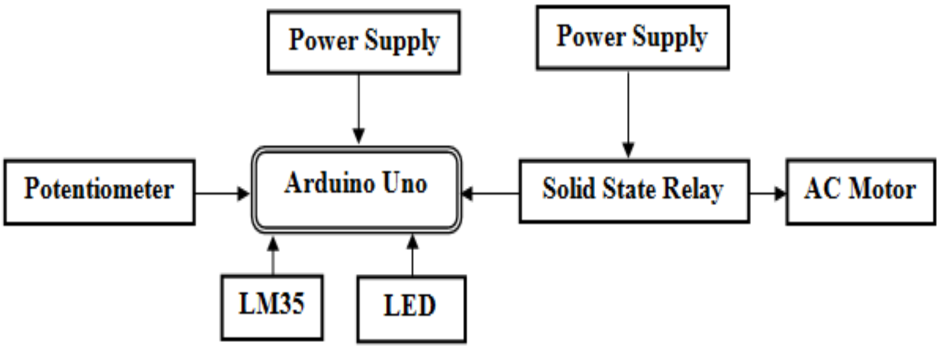
Figure 2: Final Guard Mounting 3D Ilustrations with 2D Drawing and Construction



SYSTEM DESIGN – MECHATRONICS ENGINEERING

The electronically controlled part of the system has two main components. The speed controller and the temperature monitor. However, it is inter-related, if the temperature of the motor rises over a certain value than the whole system shuts down and the speed controller will not function until the sensor senses the cooling of the motor to a temperature acceptable for operation. Below is the block diagram showing the electronic system and its inter-relation.

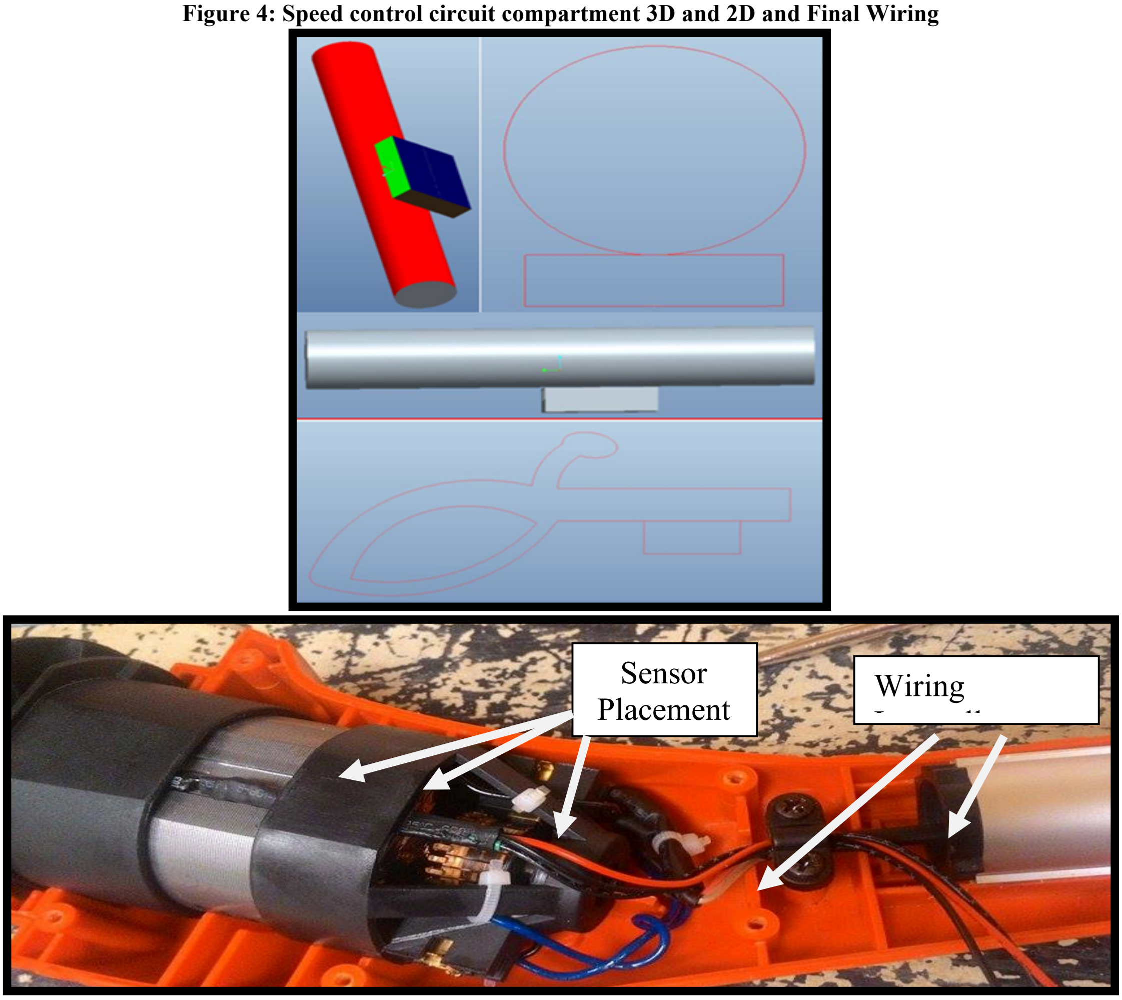
Figure 3: Electronic System Inter-relation Block Diagram



The control system comprises of the potentiometer being used as the control knob, the Arduino as the controller unit and the solid state relay is cutting down current by switching on and off at a high speed. Starting from the potentiometer, it has three pins, one connected to 5 volts and the other connected to ground, creating a difference of 5 volts. The center pin is connected to the inner wiper which is turning clockwise or anticlockwise changing the resistance as it turns. So the 5 volts is divided between 1023 steps of the turn. This is read and recorded into the Arduino analogue pin A0 and mapped to the digital pin 9 between 0 to 255 and outputted to the solid state relay which will send the increasing or decreasing signal to the motor in a fast switching manner. In addition to this, the LM35 sensor is connected to the motor to monitor the temperature. This is also connected to the analogue A1 pin into of the Arduino and grounded and whatever voltage is recorded is calculated into a temperature value using coding and a constant multiplier. This recorded value is compared to a set value of 95 degrees, depending on the condition, the system will continue to work or trip off.

As the image below shows, the sensor is wrapped in shrink tube to protect the wiring from melting and fitted between the rubber casing and the motor itself. This way, the sensor is in direct contact and can record the temperature reading as accuratly as possible.

Figure 4: Speed control circuit compartment 3D and 2D and Final Wiring



DRIVE MOTOR ENHANCEMENT

Considering the following objective of implementing speed control, the motor was replaced with with more a compatible version to the application. Number of poles in the winding combined with the frequency (Hz) helped determine the revolution of the motor. All four pole motors run at the same speed and so will all six pole motors. The fundamental calculation is shown below. Number of cycles of the frequency (Hz) multiplied with one minute (60sec) multiplied by two for both sides of the cycle finally divided by the number of poles.

Synchronous Speed = (Freq * 120)/No. of Poles

Power consumption was minimized. The control system includes three main parts that need power supply.

One Arduino Uno $= Current \times Voltage$
$= 46.5mA \times 5V (According\ to\ the\ datasheet)$
$= 232.5mW$

One AC Motor $= Current \times Voltage$
$= 300W$ (According to the datasheet)

One SS Relay $= Current \times Voltage$
$= 25A \times 5V$ (According to the datasheet)
$= 125W$

MACHINE PROGRAMMING & CODING

There are two main inputs (Sensor and Potentiometer) and two outputs (LEDs and Motor) in this system. As shown in the section of the code below, the potentiometer input to the Arduino is set to the analogue pin A0 while the Motor pin is set to the digital write pin D9. And it is initiated as value 0. This plays a key role every time the system restarts. It will initiate from zero instead of picking up where it left off.

```
int potpin = A0;
int motorpin=9;
int potvalue=0;
int motorvalue=0;
```

This section of the code is setting the pin 4 and pin 5 as output, later to be used for the LEDs. Serial begin 9600 is a debug function to help the programmer spot errors.

```
void setup() {
  // setup code to run:
Serial.begin(9600);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
}
```

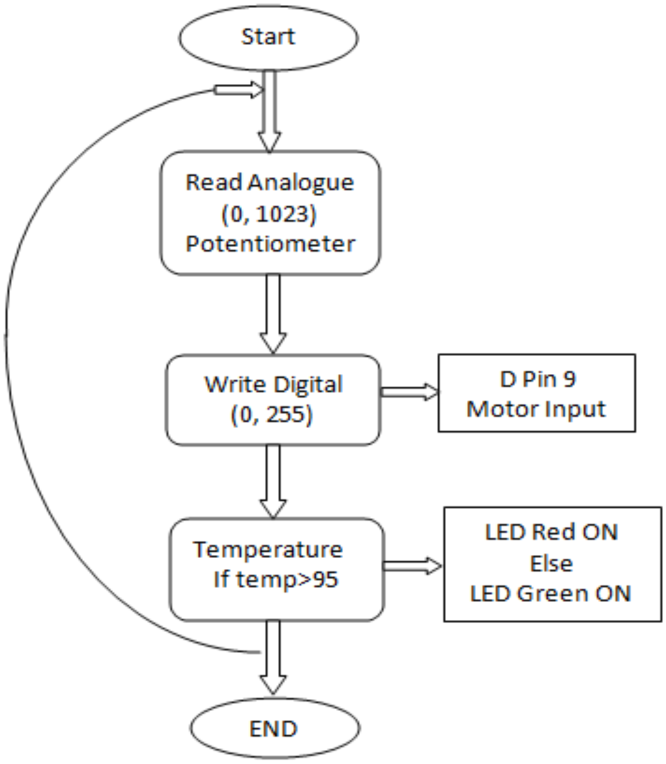
Void loop will start the infinite looping of the system. The next line is getting the reading from the temperature sensor from the analogue A1 pin and storing it as “temp”. This “temp” value is stored in “tempa” after multiplying it with a set value of 0.38838125 to get the voltage reading to temperature conversion. Last line is storing the potentiometer value from the analogue pin A0 which was previously stored in ‘potpin’.

```
void loop() {

  int temp= analogRead(A1);
  int tempa= temp*0.38828125;
  potvalue = analogRead(potpin);
```

The flowchart below shows the overall system process and illustrates the system development and process as it summarizes the algorithms used in the coding of the system.

Figure 5: Summary of Algorithms



The input of an analog pin is read and mapped the result to a range from 0 to 255 and then uses the result to set the digital output to the motor.

Motorvalue = map(potvalue, 0,1023, 0,255);

In this program, two assignments were declared, analog 0 for the potentiomer and digital 9 for the motor. In the main loop, *PotValue* stores raw analog values coming from the potentiometer. Since the Arduino has an analogRead resolution of 0 – 1023 and an analogWrite resolution of 0 – 255 the analog input data needs to be scaled before using it to control the speed. This is where the map() function comes in. O*utputValue* is assigned the value that has been scaled from the potentiometer input through the map() function. The new values are then output to the PMW Pin 9. This leads to the speed changing as you turn the speed knob.

FINAL DESIGN OUTPUT

The components selected play an important role on the products performance. There are seven keys components for this research project as listed below. The Arduino Uno was selected over the PIC1F for its easy interface with other components such as the solid state relay and the LM35 sensor and importantly it is easily programmable in comparison to the PIC. The LEDs are basic and not much selection process has taken place. However the temperature sensing was trailed with thermistor and other thermostats. The accuracy level of the reading and the interfacing was more complicated and needed more calibration than the LM35. The potentiometer was selected after trying the three point switch; the potentiometer has gradual increase and decrease rather than limiting to three set speeds.

The Pin configuration for wiring to the Arduino is listed below. The LM35 signal is connected to Analogue 1 (A1), while the Vcc is connected to Voltage in pin (Vin) and one end grounded (GND). The potentiometer is connected to Analogue 0 (A0), while one end is also grounded (GND) and the other connected to 5 Volts supply (5V), The two LEDs are connected to digital 4 and 5 and grounded at the other end. The solid state relay is grounded at one end and connected to digital 9 (Pin 9 pwm). And the AC mains connected to the other ends.

Table 1: Component Pin Configuration

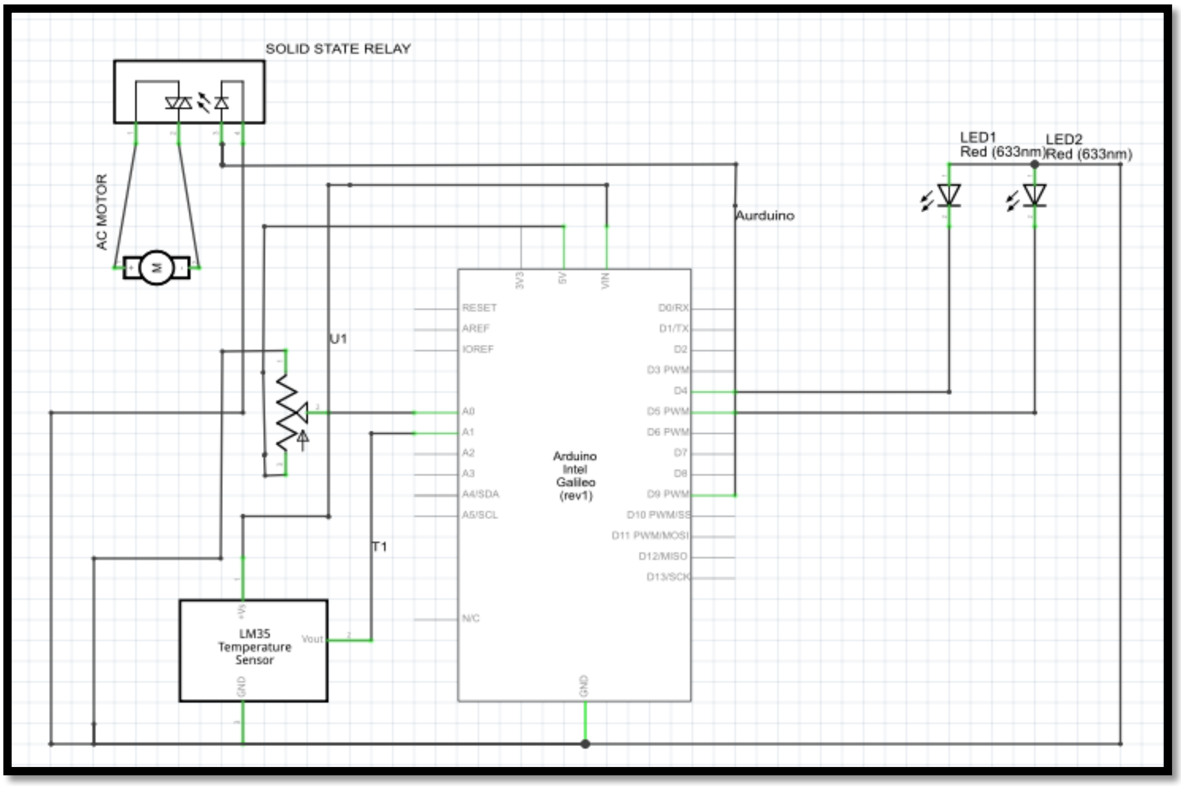
PIN	Component
A0	Potentiometer Signal Input
A1	LM35 Signal Input
D4	LED Indicator
D5	LED Indicator
D 9 (PWM)	Solid State Relay Input
Vin	LM35
5V	Potentiometer

GND	LED, LM35, Potentiometer, SSR

The schematic below shows the connection of wirings and components of the whole system. This drawing was made on 'fritzing' software.

Most of the operational issues have been eliminated but there is still one major limitation which is the consideration of the operational weight of the machine. Furthe research is expected in the material usage for construction of the machine perhaps moving into fibre glass or light weight material. This can be further developed locally in Malaysia rather than being manufactured in China.

Figure 6: Arduino interface with Relay/ Relay interface with Motor/Sensor



REFERENCES

Muhammed Naim Leman, Sharol Mohammaddan (2017). Development of Control System for Keropok Keping Drying Machine. *Matec Web of Conferences 87*, 02016, ENCON 2016.

Prithviraj R.Shetti, Ashok G.Mangave (2016). DC Motor Speed Control With Feedback Monitor Based on C# Appllication. *International Journal of Research in Engineering and Technology*, eISSN:2319-1163/pISSN:2321-7308.

Boglietti, A., Cavagnino, A., Pastorelli,M., Staton, D., and Vagati, A. (2006). Thermal Analysis of Induction and Syncronous Reluctance Motors. *IEEE Transactions on Industry Applications*, 42(3), 675-680.

Chen, Y., Sun, S. and Liu, Y. (2007). Numerical simulation of the mechanical properties and failure of heterogeneous elasto-plastic materials. *Tsinghua Science and Technology*, 12(5), 527-532.

Ding, Y., Lan, H., Hong, J. and Wu, D. (2004). An integrated manufacturing system for rapid tooling based on rapid prototyping. *Robotics and Computer-Integrated Manufacturing*, 20(4), 281-288.

Ibrahim Yildiz (2016). A Passive Stewart Platform Based Joystick to Control Spatially Moving Objects. *ACHI 2016 : The Ninth International Conference on Advances in Computer-Human Interaction*.

Xianjun, S., Lingshun, i. and Shutuan, h. (2012). Research for Digital DC PWM Controlling System. *Energy Procedia*, 16, 1547-1552