

CM10194 Computer Systems Architecture 1 - Coursework 2

Overview

For this coursework we were given the task of creating a prototype solution for a real world problem that uses at least 3 input sources and 1 meaningful output.

For my project I decided to base my Arduino on trying to prevent break-ins and burglary, whether that be for a whole building or for a room. My coursework could also be used within families to stop uninvited people from entering each other's room i.e. Siblings.

Using this 'burglary' problem I decided to create a prototype of a laser security system, whereby the system can be turned on via button or when there is a lack of light within the room. The system is then triggered when there is a break between the laser and the LDR. When the system is triggered sound is outputted as well as flashing LEDs.

My project can be seen in the appendix at A,B and C. The schematic can be seen at appendix D. My video will demonstrate the features of my project. In my schematic as there was no symbol for the laser I had to use the diode symbol for it, but I did label it laser for you reference.

Inputs & Outputs

For my inputs I decided to use an LDR, Phototransistor and a button. I decided to use an LDR as the sensor for my laser, so when the security system was on the laser was pointed towards the LDR, the values of the LDR were constantly being recorded and as soon as the laser was not touching the sensor i.e. there was a break in the laser, the alarm would be triggered.

The phototransistor was used to automatically turn on the security system, when the system was off values were being read from the phototransistor constantly, the values recorded were the amount of light that is in the room. When the light in the room dropped below a certain threshold the security system would turn on, the threshold value was such that the security system would only turn on if the room was almost in complete darkness. I decided to implement this feature incase the user forgot to turn the alarm on when leaving their room/house the security system would automatically turn on if no lights are on in that area.

The last input was a button, this was to simply turn the security system on/off or to turn the alarm off when it was triggered.

For the outputs I used a 5V laser, some LEDs and a piezo. The laser is turned on when the security system is on, it is pointed towards the LDR as mentioned above, when anything crosses the laser when on the alarm is triggered. It is turned off via button and only via the button. I used three LEDs, the green one is used to indicate the system is on, the white and blue LEDs, are turned on when the alarm is triggered and they can only be turned off via the button. The piezo is used as a sound output for when the alarm is triggered, it outputs a siren noise and can also only be turned off via the button.

My Code

The first part of my code is initializing variables and setting the pins for each of my inputs and outputs. Unfortunately when I was building my prototype I found that my LEDs only light up properly in certain pins in the board.

Within the setup function I was setting the pins and the parts to Output and Input using pinMode. I also set the values for the variables of alarmStatus and buttonStatus to 0, this is to ensure they are both off when the system is first plugged in. There is also an interrupt for the button, so every time the button is pressed the interrupt happens, the interrupt immediately runs the function checkStatus. This function is used to check the status of the alarm at the time of the button press and acts accordingly as can be seen in my code. For example if the alarm is off and the button is pressed, the alarm status is then set to 1 to turn the alarm on. It also adjusts the buttonStatus, but when the alarm is off and the button is pressed the buttonStatus is still set to 0, meaning it has not been pressed. This was done as I encountered a problem within my prototype, when the alarm was turned on via the lack of light in the room, and if it was then triggered and turned off via the button, the alarm could not be turned back on via the button unless the system was reset by uploading the code again. So to prevent this from happening I had to set the buttonStatus to 0 when it was pressed when the alarm was off.

Within the main loop function the code constantly checks the alarm status and acts accordingly. When the alarmStatus is 0 it turns all the LEDs and laser off. If the alarm status is 1 the code turns on the laser and the green LED, it then proceeds to keep checking the value of the LDR via the alarmTriggered function. The alarm triggered function contains an if statement where if the reading from the LDR is below 600, meaning there has been a break in the laser and LDR, and the alarmStatus is 1 it sets the alarmStatus to 2. If the alarmStatus is 2, the loop runs a series of code which outputs a siren sound on the piezo and turns the blue and white LEDs on. The alarm keeps repeating until the button has been pressed to turn the alarm off.

Also within the loop there is a function that is run if the alarm is not already on or triggered, this function is called checkLight. The function reads the value from the phototransistor and decides whether to turn the alarm on depending on the threshold value, the threshold value I have set is for when the alarm is in a dark environment it turns on. The code within this functions sets the alarmStatus to 2 when the phototransistor is reading a value below 5.

Improvements

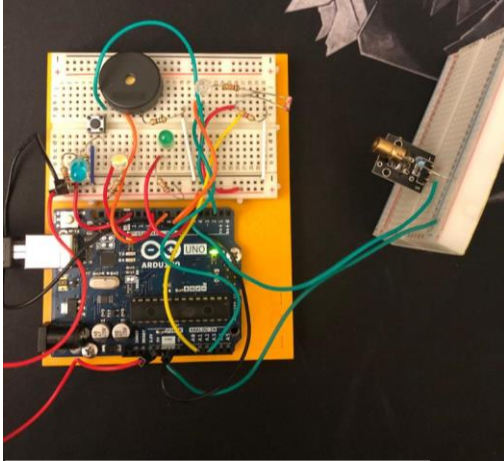
For this prototype security system, although the functionality is working perfectly fine, there could be some improvements.

My first improvement would be to add a remote connection feature using an app system on a mobile phone. This app could be used to turn the alarm on/off as well as to check if it has been triggered. I would also add a feature whereby when the alarm is triggered the user will also get an alert on their phone.

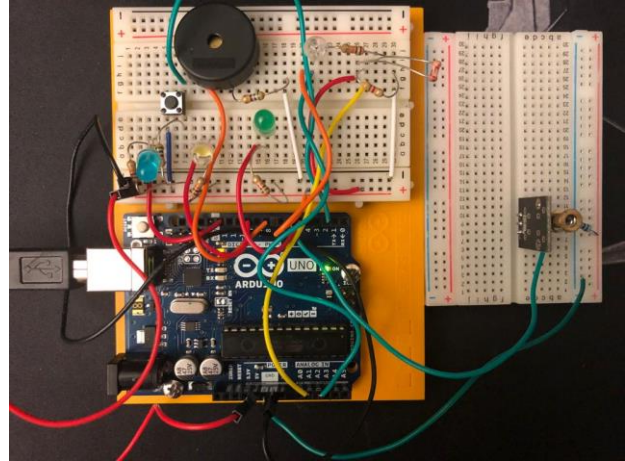
Another improvement to the alert system I would incorporate would be to allow the user to add a second user, not to enable or disable the alarm, but the second user would also receive an alert via SMS when the alarm is triggered

My final improvement would be to add a passcode system. The passcode would be used to enable/disable the alarm, this would ensure only people who know the passcode can access the security system. I would have tried to incorporate this into my prototype, but as the LCD screen requires several pins on the Arduino this was not possible.

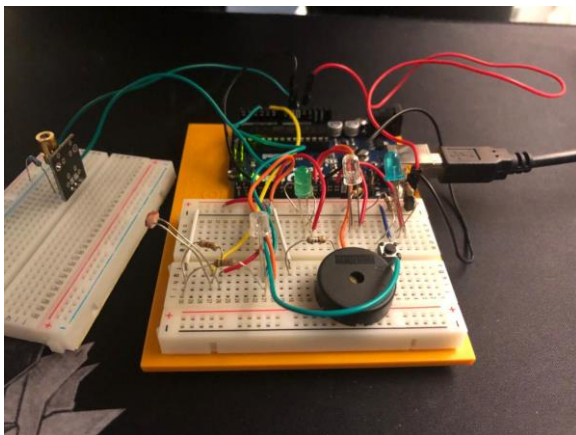
Appendix



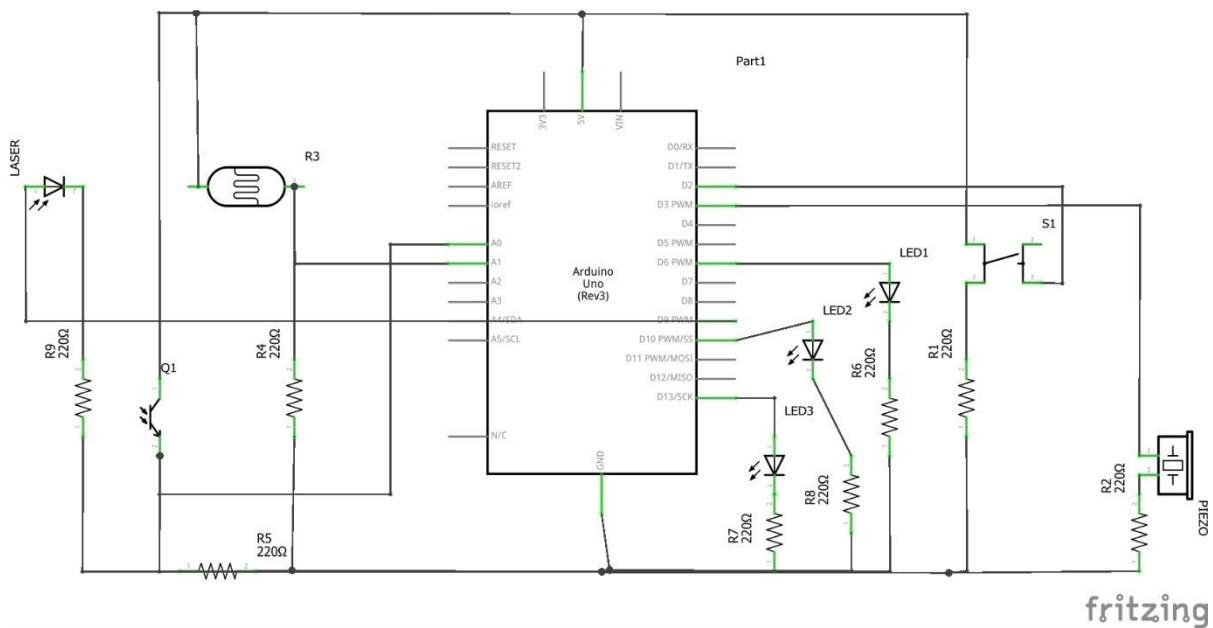
Appendix A



Appendix B



Appendix C



Appendix D