2017

LARM

SIMULERING AV ARDUINO

KRISTOFER@SOMLIN.SE

Innehåll

Kristofer Somlin
Mölk Utbildning

Syfte

Syftet med detta arbetet var att försöka skapa en start till ett larmsystem.

Kristofer Somlin
Mölk Utbildning

Metod

Då detta arbete går ut på att skapa en simulering av ett projekt. Eftersom mina tidigare projekt inte gick att simulera med tinkercad så fick jag skapa mig ett nytt.

Jag hade många olika idéer när jag började men eftersom jag tidigare varit intresserad av att bygga ett larm tänkte jag att detta var ett lysande tillfälle att göra det. Jag började med att sätta upp en knapp med hjälp av intern pullup, kopplade sedan en led för att kunna få ut information utan att använda den seriella monitorn. Efter detta adderade jag ytterligare 2 knappar och en led till. En knapp var tänk att ersätta en blivande sensor, t.ex. en rörelsesensor. De andra knapparna var istället för knappsats, den ena knappen ger ifrån sig rätt kod för att aktivera eller avaktivera larmet. Den andra ger ifrån sig fel kod. Den gröna led lampan var tänkt att lysa när larmet är aktiverat eller blinka x antal gånger när man slagit in fel kod. Den röda lampan är tänkt istället för siren eller liknande, om någon bryter den blivande rörelsesensorn och larmet är aktiverat kommer denna blinka tills man slagit in den korrekta koden. Då tinkercad ofta buggade för mig så började jag programmera i Atom för att sedan provköra koden i Arduino IDE. Detta för att kontrollera så att koden gick att kompilera. När det var gjort så kopieras koden in i tinkercad och kan oftast simuleras.

Resultat

Genom att använda sig av Arduino IDE för att testa koden innan simulering minimeras tiden som lägg på felsökning. Ska man bygga ett larm så borde detta byggas på ett realtidssystem för att få det så säkert som möjligt och minimerar buggar.

Kristofer Somlin
Mölk Utbildning

Diskussion / Analys

För ett larmsystem vore det bästa att använda sig av en lösning med ett realtidsoperativsystem. Då man vill kunna hantera flera saker samtidigt så är inte en Arduino den optimala hårdvaran. Jag har även kommit fram till att det bästa och smidigaste sättet för mig att jobba med tinkercad är att först skriva koden i Atom, sedan klistra in den i Arduino IDE för att kompilera och kontrollera att det inte är några fel. När det är färdig klistra in det i tinkercad.com och starta simulering. Det har tagit lärdom av är att man först ska undersöka vilka olika simuleringsverktyg det finns och jämföra dessa mot varandra. Tinkercad.com har varit ett bra verktyg för att vara gratis, dock mycket buggar så man blir lätt lite frustrerad.

Kristofer Somlin
Mölk Utbildning

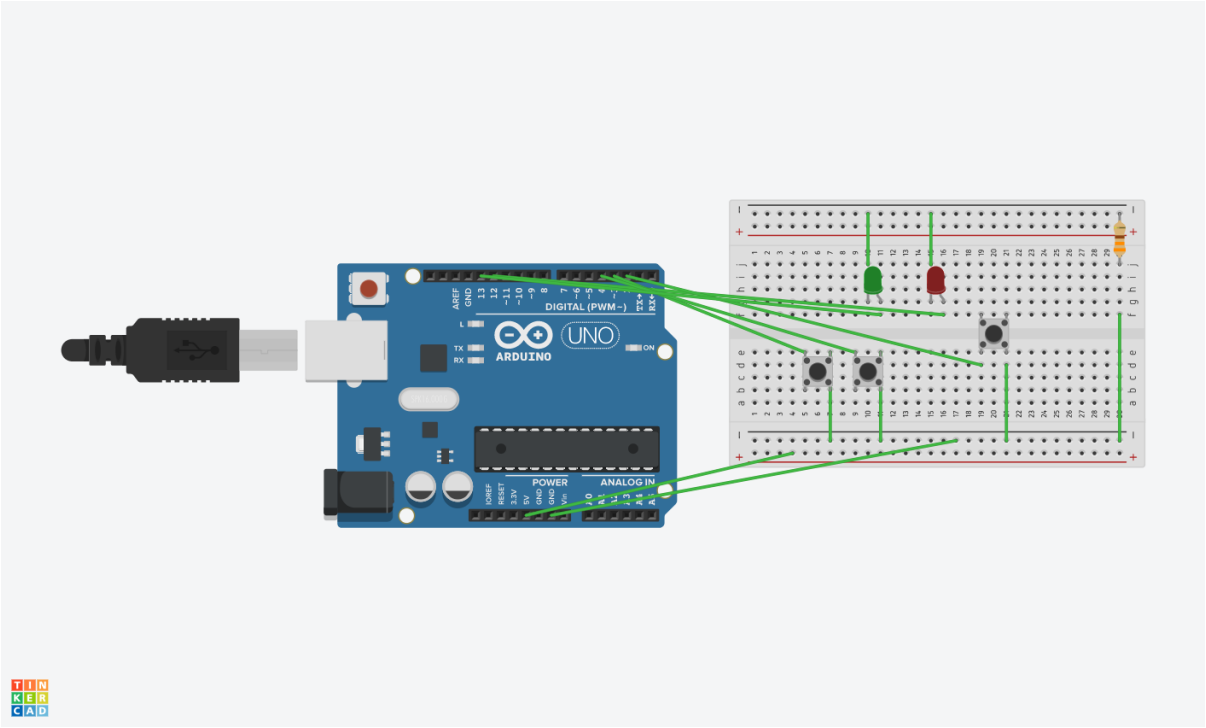
Slutsats

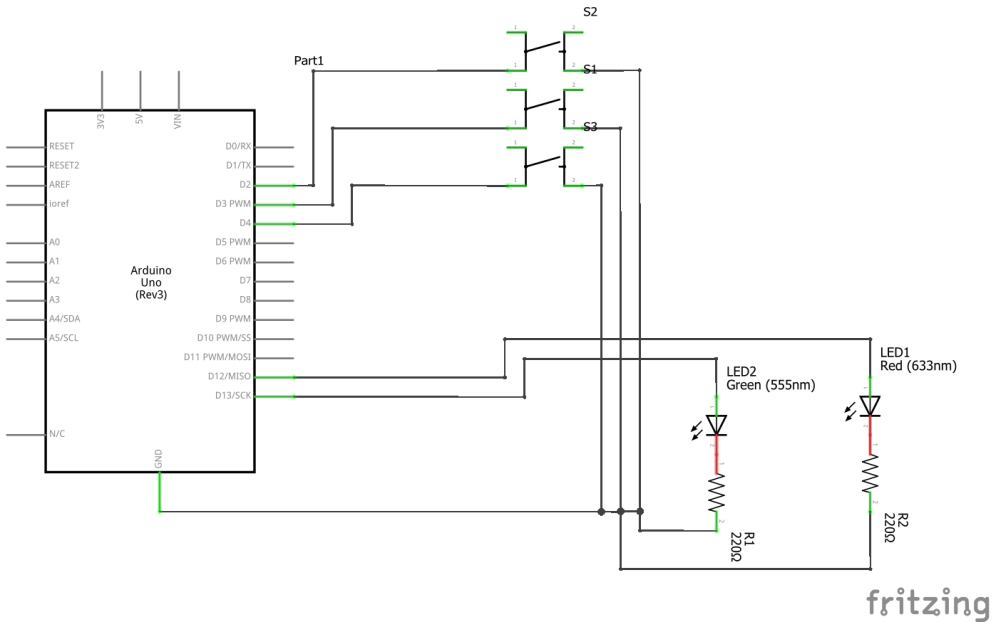
Till nästa gång man skall göra ett simuleringsarbete kolla upp mer vilka mjukvaror man kan använda sig av i stället för tinkercad. Kanske kolla upp en simulator för ATmega 328p eller liknande. Men på grund av tidsbrist så har det ej blivit så denna gång. Tycker dock det är bra att få upp ögonen för simulering då det är ett väldigt bra verktyg. Kan vara smidigt för att kunna räkna ut hur man ska koppla in saker innan man gör det på en riktigt breadboard eller till och med kanske löder.

Kristofer Somlin
Mölk Utbildning

Bilagor

https://www.tinkercad.com/things/e6VLJJ4OEr9-smashing-bruticus/editel?sharecode=ExXKxz7MPurKFwRaji7BduBXnXkdYHsCV8ceJyR9V3o=





Kristofer Somlin
Mölk Utbildning

```
int GREEN_LED=13,RED_LED=12;
int larm_status=0,thief_detected=0;

void setup() {
 Serial.begin(9600);

 // WILL BE USED INSTEAD OF SENSOR
 pinMode(2, INPUT_PULLUP);

 // WILL BE USED INSTEAD OF USING CODE TO TURN ON OR OFF THE ALARM. (THIS IS THE CORRECT CODE)
 pinMode(3, INPUT_PULLUP);

 // WILL BE USED INSTEAD OF USING CODE TO TURN ON OR OFF THE ALARM. (THIS IS WRONG CODE)
 pinMode(4, INPUT_PULLUP);

 // WILL BE GREEN IF THE ALARM IS ON, BLINK 5 TIMES IF THE CODE IS WRONG
 pinMode(GREEN_LED, OUTPUT);

 // WILL BLINK IF SOMEONE/SOMETHING IS IN THE HOUSE WITHOUT TURNING THE ALARM OFF.
 pinMode(RED_LED, OUTPUT);
}

/**
 * [blink_light description]
 * function to blink x-output y-times.
 * @param light [will use this port to toggle high and low]
 * @param num   [how many times this will be repeated]
 */

void blink_light(int light,int num) {
 for(;num>0;num--) {
   digitalWrite(light,HIGH);
   delay(100);
   digitalWrite(light,LOW);
   delay(100);
 }
}

/**
 * [check_code description]
 * function to compare 2 codes with each other
 * if they match it will invert larm_status
 * @param correct_code [this input should be the correct code for the larm]
 * @param compare_code [this input is the code entered with the numpad]
 * @param larm_status  [a pointer to larm_status so it can be toggled]
 */

void check_code(int *correct_code, int *compare_code, int *larm_status) {
 int i=0;
 for(;*correct_code==*compare_code;i++,*correct_code++,*compare_code++) {  }
 if(i==4) {
   *larm_status=!*larm_status;
   thief_detected=0;
 } else {
   blink_light(GREEN_LED,10);
 }
 delay(500);
}

/**
 * [larm_triggered description]
 * this function is used when sensor is triggered and the larm is on]
 */
void larm_triggered() {
 digitalWrite(GREEN_LED,LOW);
 thief_detected=1;
}

void loop() {
 int larm_code[4]={1,3,3,7};
 int right_code[4]={1,3,3,7};
 int wrong_code[4]={5,4,1,7};
 // WILL SET THE LARM OFF. INSTEAD OF USING SENSOR
```


Kristofer Somlin
Mölk Utbildning

```
 if(!(digitalRead(2)) && larm_status) {
   larm_triggered();
   Serial.println("SENSOR ACTIVATED");
 }

 if(!(digitalRead(3))) {
   check_code(larm_code,right_code,&larm_status);
   Serial.println("CODE: 1337");
 }

 if(!(digitalRead(4))) {
   check_code(larm_code,wrong_code,&larm_status);
   Serial.println("CODE: 5417");
 }

 if(!(thief_detected)) {
   if(larm_status) {
     digitalWrite(GREEN_LED,HIGH);
   }

   else if (!(larm_status))  {
     digitalWrite(GREEN_LED,LOW);
   }
 }
  else {
   Serial.println("THIEF DETECTED");
   blink_light(RED_LED,1);
 }
 delay(10);
}
```


Kristofer Somlin
Mölk Utbildning