

The mini project

- Session October 13 team presents mini project proposal to coach (Location to be arranged)
- Sessions October 18, 20, 25 no lectures, but support in and around e-labs for mini project development
- Session October 27 (LG 1.105) team presents mini project result to coach (demo + mini poster) (Location e-labs)

The mini project

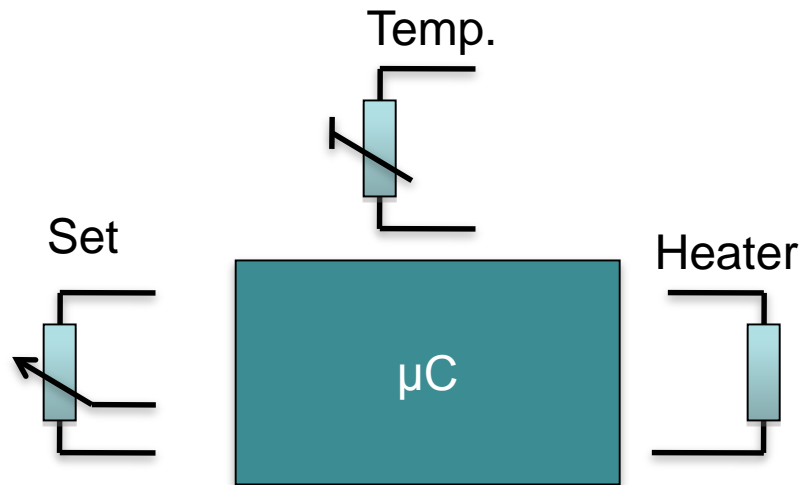
- Proposal
 - Indicate context
 - What should it do
 - Input/output -> sensors/actuators

- You are free to choose your project, but:
 - Use Arduino if possible
 - Use at least one sensor and one actuator
 - Report on design choices (transistor, configuration, etc.)
 - Draw schematics
 - Present it as a **prototype** (user experience, packaging) !



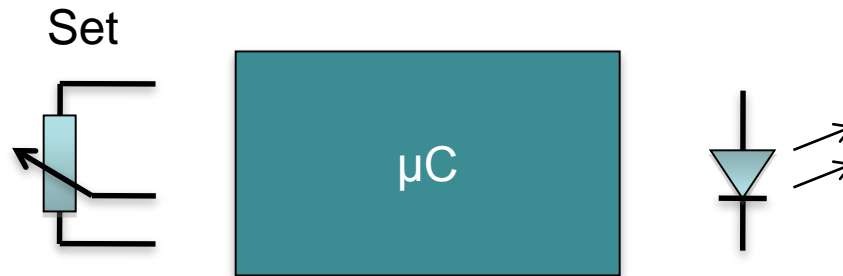
Suggestions

- Make the central heating system with Arduino



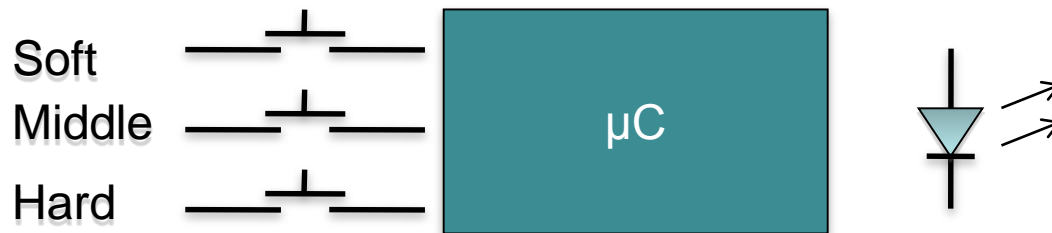
Suggestions

- Dim a LED
- Make application / context



Suggestions

- Egg timer



Spaghetti in code, schematics and circuits



Code:

- Use comments `/* like this */` or `// this`
- Use functional variable names
- Use functions

Schematics:

- Draw the complete circuit (correctly!!)
(component names, values)

Circuit (soldered or breadboard)

- Should be structured
- Use color coding in wires
- Think about component placement

Otherwise:

- Debugging becomes difficult
- You lack evidence in your report

Code

```
const int Baudrate = 19200;
const int Analog_Input_Pin_A = 0; // Sensor A is connected to this pin
const int Analog_Input_Pin_B = 1; // Sensor A is connected to this pin

void setup()
{
  Serial.begin(Baudrate);          // For sending data to the computer over USB
  WriteStartCommand();
  InitialiseSensors(Analog_Input_Pin_A, Analog_Input_Pin_B);
  InitialiseTimers();
}

void loop() {
  Read_Sensors_and_Write_Data();
  if (SerialCommand()) {
    Do_Something_With_Command();
  }
}

/* Implementation of subroutines:      */
void WriteStartCommand(void) {
```

Done Saving.

Binary sketch size: 1372 bytes (of a 30720 byte maximum)

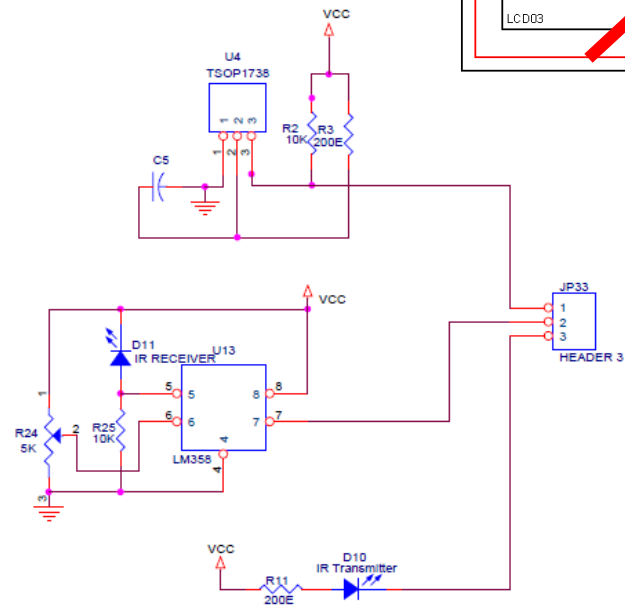
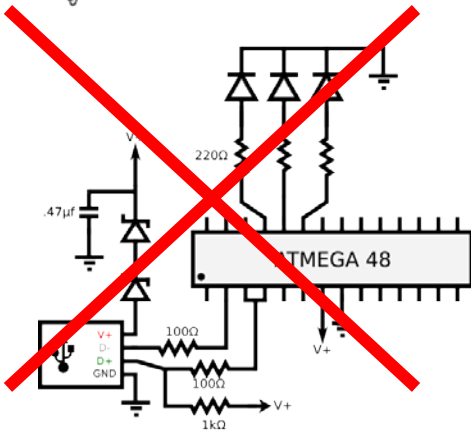
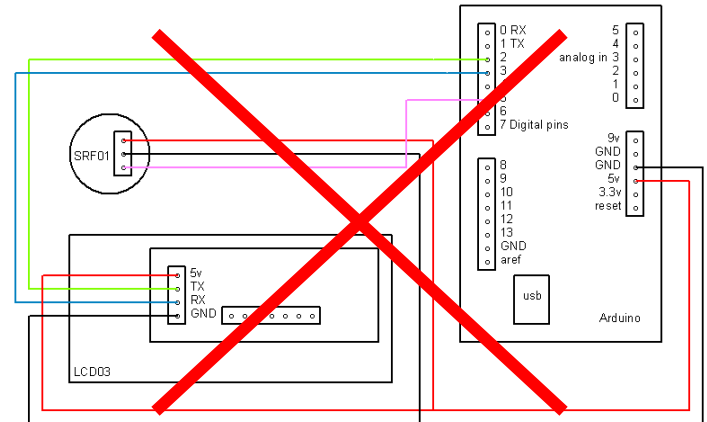
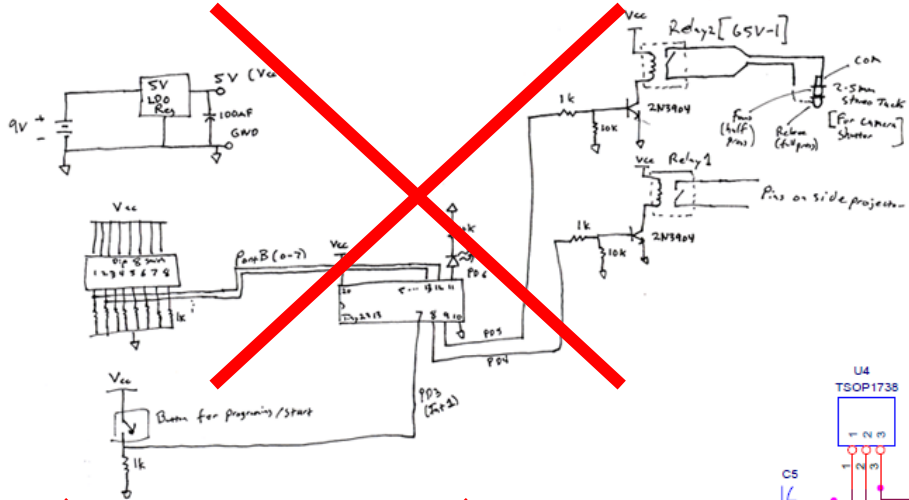
20

Code:

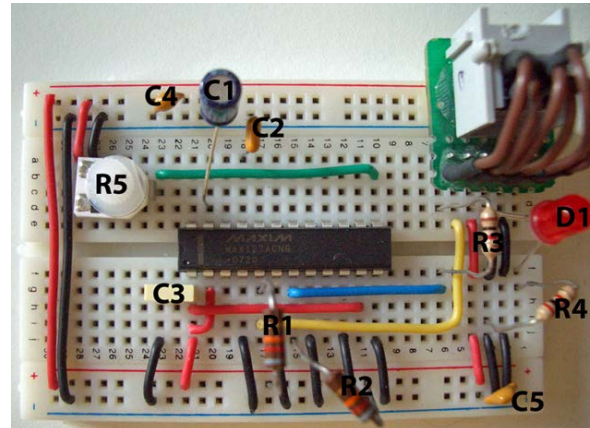
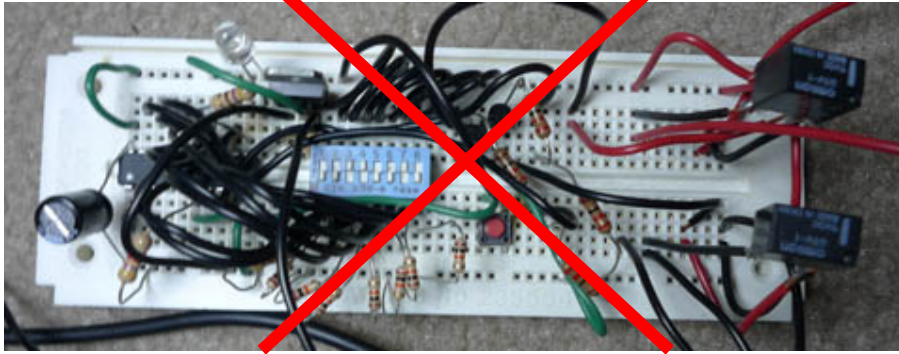
- Clear variable names
- Proper use of local and global variables
- Functions `setup()` and `loop()` are self-explaining
- Proper function names
- Comments (Header)

- Examples
 - [Bad example](#)
 - [Good example](#)

Schematics



Circuits

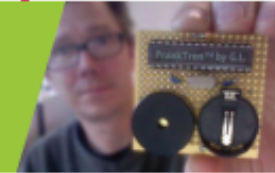


Poster

- Explanation
- Schematic
- Code
- Pictures

PrankTron: your nightmare

Geert Langerels (B24-1)



Introduction

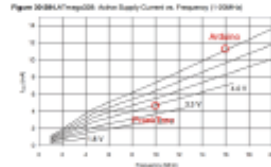
PrankTron is a small electronic module that selects randomly one out of four sounds and bothers you with that at irregular times between 10 minutes and 1.5 hours. It is small enough to fit in any obscure location, for example in the board room of a study association. PrankTron is inspired by the commercially available Annoy-a-Trons¹.

Specifications

- Small size (smaller than Arduino)
- Low power (should stay active for 10 days)
- Several sounds

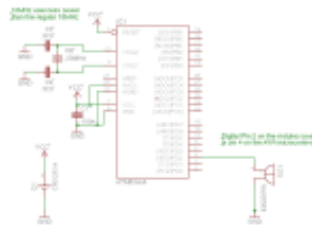
Electronics

The only thing needed for a PrankTron is a buzzer. A 5V buzzer can be connected to an Arduino board directly. However, to make a low-power version, the Atmel microcontroller was taken off the Arduino board, and placed on a piece of experimentation board. Now I could use a 3V battery and a 10MHz clock (instead of 16MHz), which gives a huge reduction of power²:



So, what is achieved is:

- ATMEGA current: 3mA
- Battery CR2032H: 240mAh
- Lifetime PrankTron: 80 hours (> 3 days)



Code

The code has two tricky parts. First of all, a function for long delays had to be made, and next, a method to deal with the 10MHz clock. Debugging was done on a normal 16MHz Arduino board with `Serial.println()` commands.

```

PrankTron_1.ino
#include <Arduino.h>
#define buzzerPin 10
#define buzzerPin2 11
#define buzzerPin3 12
#define buzzerPin4 13

void setup() {
  pinMode(buzzerPin, OUTPUT);
  pinMode(buzzerPin2, OUTPUT);
  pinMode(buzzerPin3, OUTPUT);
  pinMode(buzzerPin4, OUTPUT);
}

void loop() {
  // Randomly select one of four sounds
  int sound = random(4);
  // Delay for a while (between 10 minutes and 1.5 hours)
  long delayTime = random(5400000, 9000000);
  delay(delayTime);
  // Turn on the buzzer
  digitalWrite(buzzerPin, HIGH);
  // Delay for a while (between 10ms and 100ms)
  long delayTime2 = random(10, 100);
  delay(delayTime2);
  // Turn off the buzzer
  digitalWrite(buzzerPin, LOW);
  // Delay for a while (between 10ms and 100ms)
  long delayTime3 = random(10, 100);
  delay(delayTime3);
  // Turn on the buzzer
  digitalWrite(buzzerPin2, HIGH);
  // Delay for a while (between 10ms and 100ms)
  long delayTime4 = random(10, 100);
  delay(delayTime4);
  // Turn off the buzzer
  digitalWrite(buzzerPin2, LOW);
  // Delay for a while (between 10ms and 100ms)
  long delayTime5 = random(10, 100);
  delay(delayTime5);
  // Turn on the buzzer
  digitalWrite(buzzerPin3, HIGH);
  // Delay for a while (between 10ms and 100ms)
  long delayTime6 = random(10, 100);
  delay(delayTime6);
  // Turn off the buzzer
  digitalWrite(buzzerPin3, LOW);
  // Delay for a while (between 10ms and 100ms)
  long delayTime7 = random(10, 100);
  delay(delayTime7);
  // Turn on the buzzer
  digitalWrite(buzzerPin4, HIGH);
  // Delay for a while (between 10ms and 100ms)
  long delayTime8 = random(10, 100);
  delay(delayTime8);
  // Turn off the buzzer
  digitalWrite(buzzerPin4, LOW);
  // Delay for a while (between 10ms and 100ms)
  long delayTime9 = random(10, 100);
  delay(delayTime9);
}
    
```

Results

Tests were performed by placing a 4x4 cm² PrankTron on a piece of experimentation board under the sales counter in the Lucid boardroom in November 2011. The result was hilarious, and PrankTron was only discovered after a whole day of total confusion.

References

- ¹The ThinkGeek Annoy-a-Tron <http://www.thinkgeek.com/product/8c52/>
- ²Atmel ATMEGA328P datasheet http://www.atmel.com/dyn/resources/prod_documents/doc8271.pdf