The mini project

- Session May 27 team presents mini project proposal (PAV A12a Group1 / PAV A12a Group2)

- Session June 1 no lecture, but support in and around e-lab for mini project development

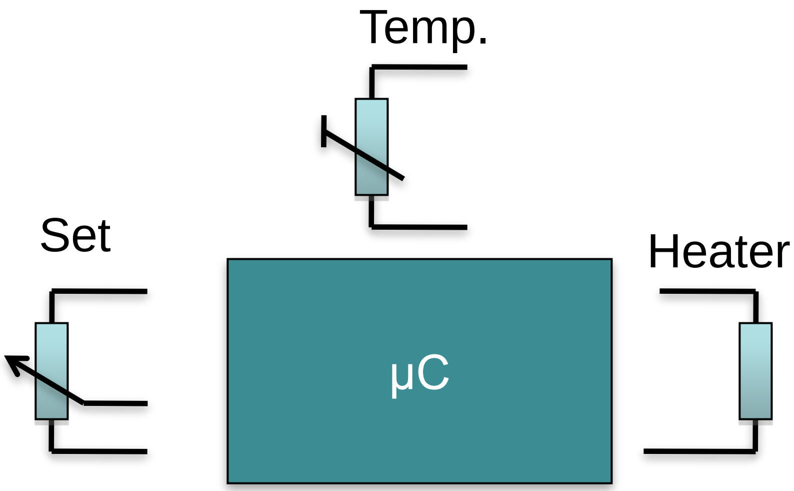
- Session June 8 (LG 1.105) team presents mini project result (demo + mini poster)

The mini project

- 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) !

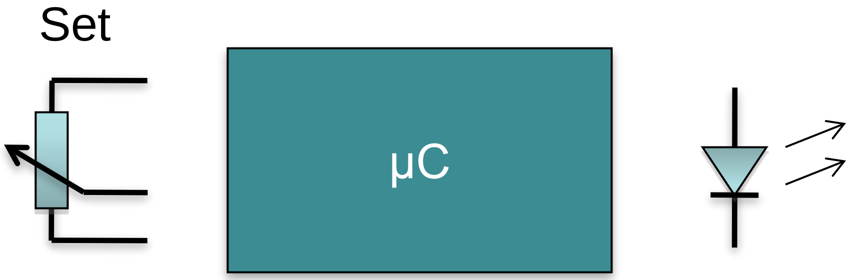
Suggestions

- Make the central heating system with Arduino



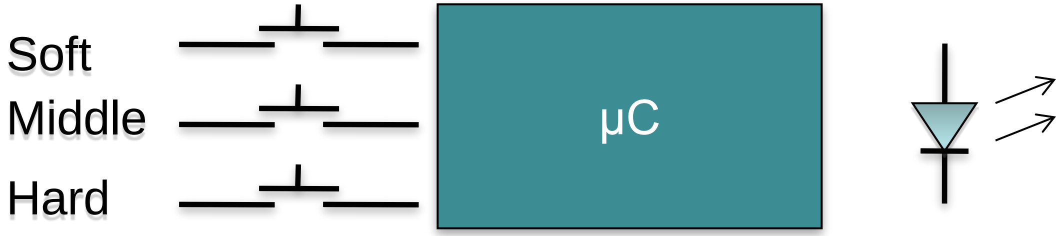
Suggestions

- Dim a LED
- Make application / context



Suggestions

- Egg timer



Spaghetti in code, schematics and circuits

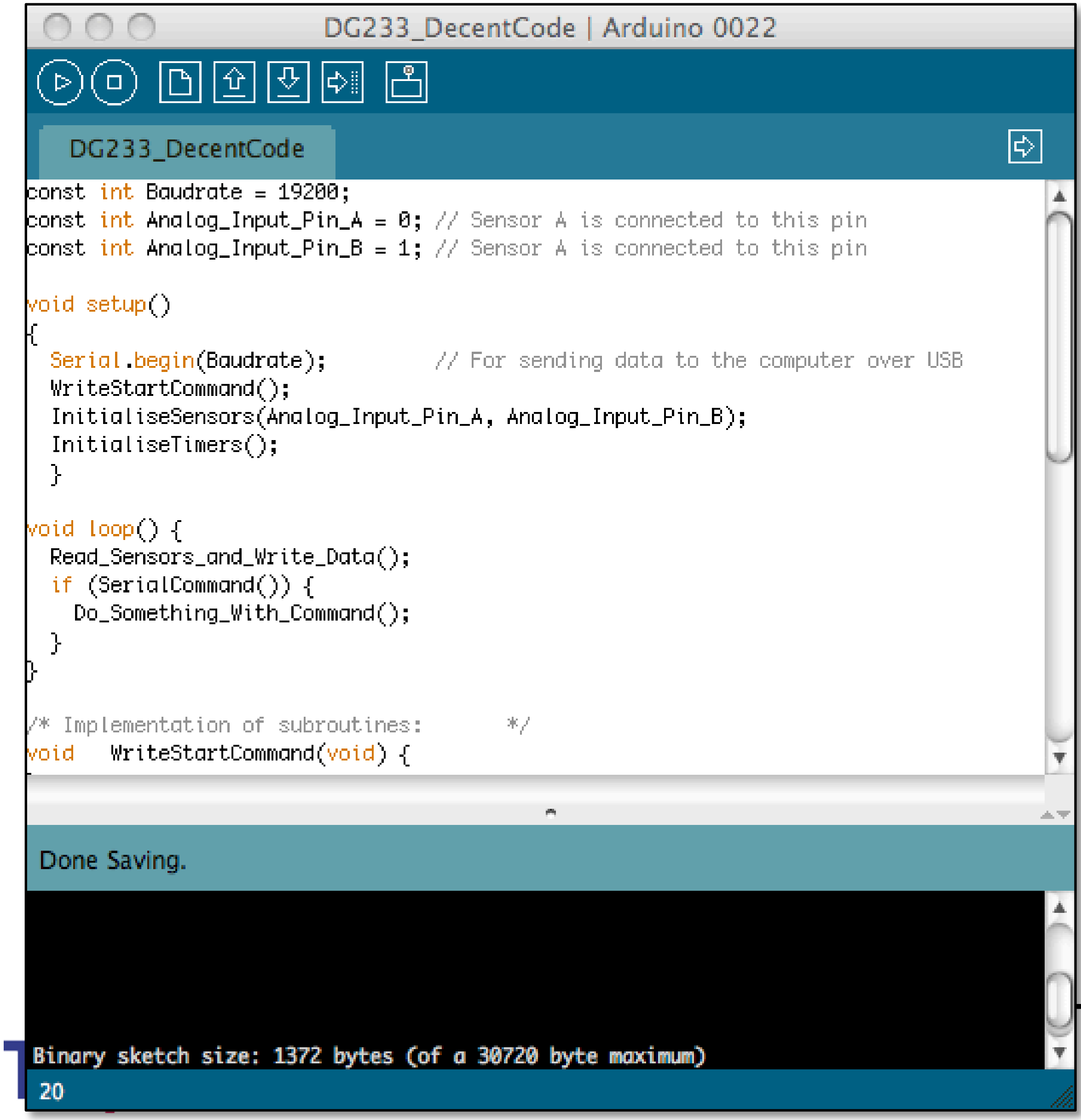
Code:
- Use comments /* like this */ or // this
- Use functional variable names
- Use functions

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)
```

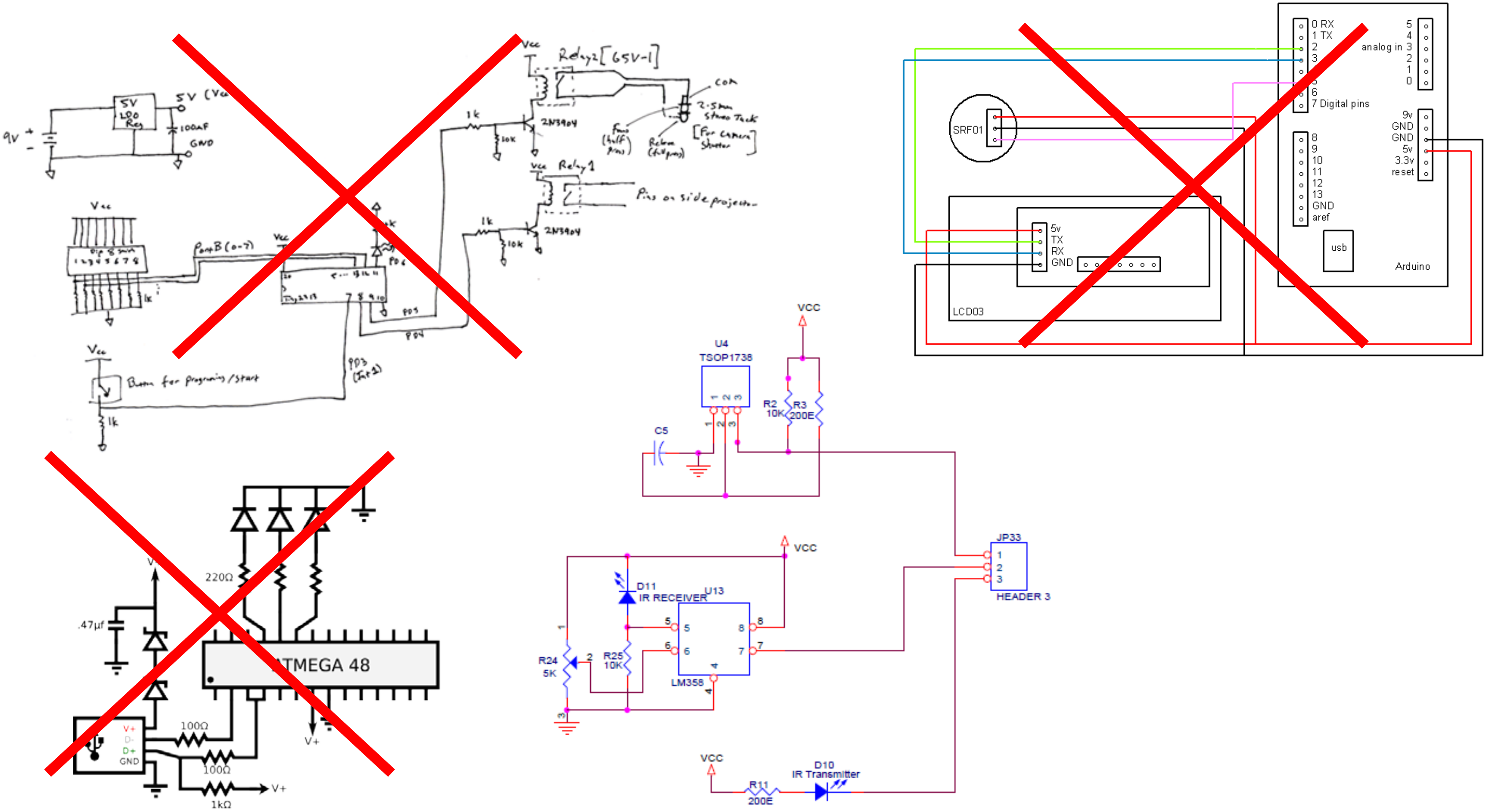
20

Code:

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

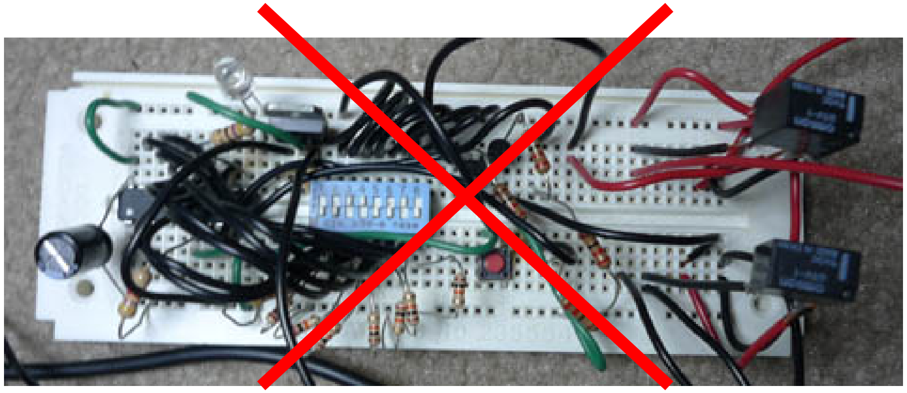
- Examples
  - Bad example
  - Good example

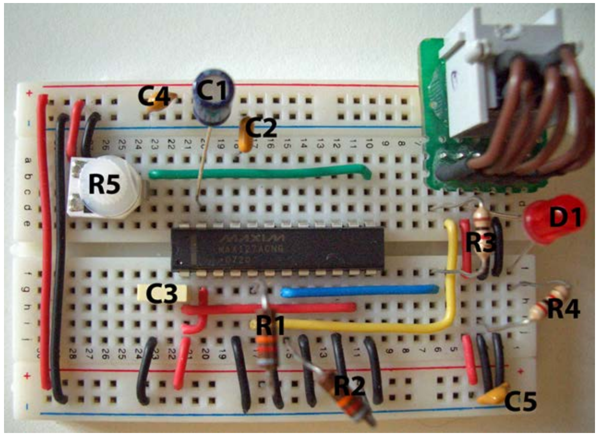
Schematics



Nov, 2014

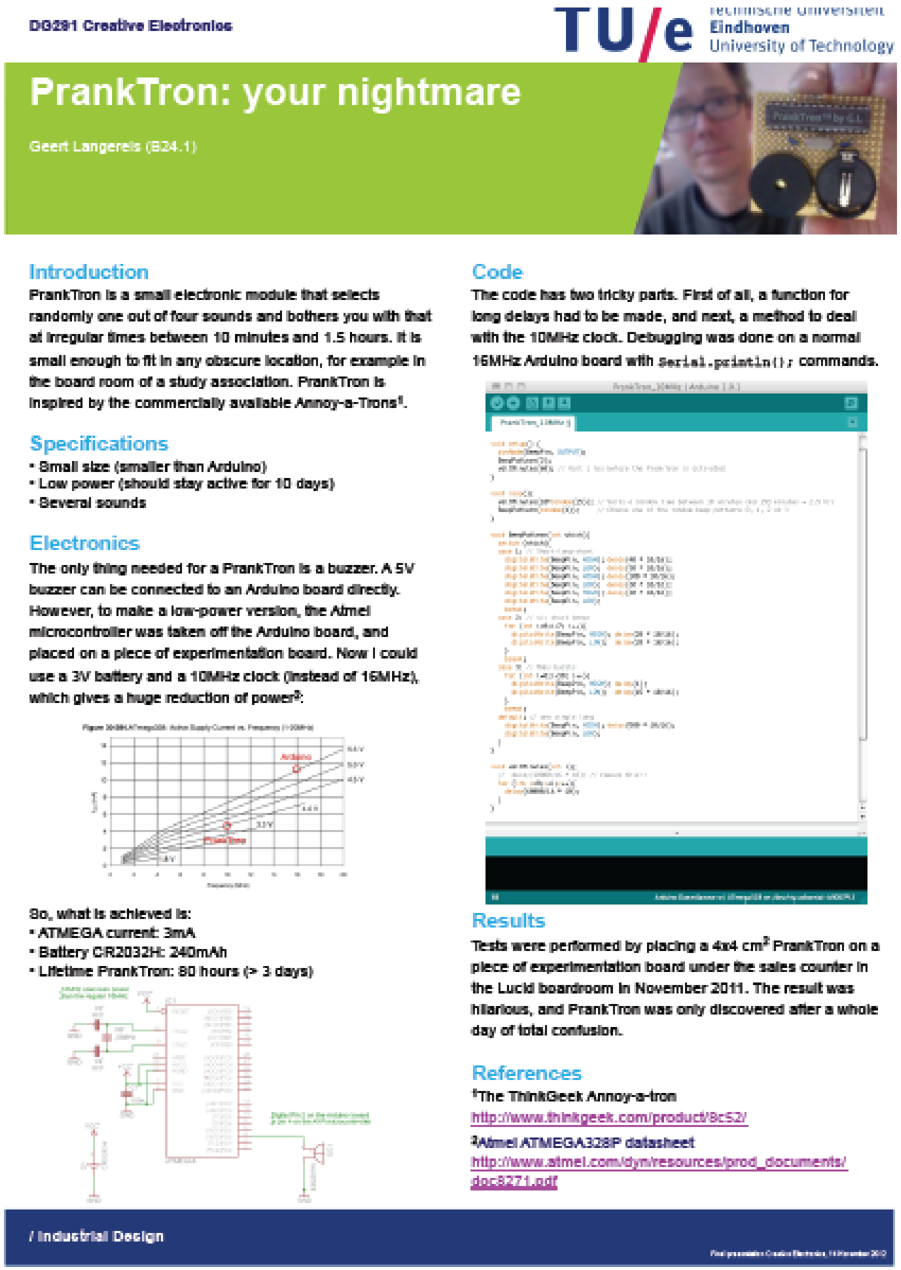
Circuits





Poster

- Explanation
- Schematic
- Code
- Pictures



DG281 Creative Electronics

PrankTron: your nightmare

Geert Langereis (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-Trons[1].

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 power[2]:

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.

Results

Tests were performed by placing a 4x4 cm^2 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

[1] The ThinkGeek Annoy-a-tron
http://www.thinkgeek.com/product/8c52/

[2] Atmel ATMEGA328P datasheet
http://www.atmel.com/dyn/resources/prod_documents/doc8271.pdf

/ Industrial Design

Resources

- Look at wiki page:
  - http://wiki.id.tue.nl/ce/AssignmentDescription (and checkout the “Resources” section)

- Installation
  - http://arduino.cc/en/Guide/Windows
  - http://arduino.cc/en/Guide/MacOSX