Tennis scoreboard with ESP32 WROOM

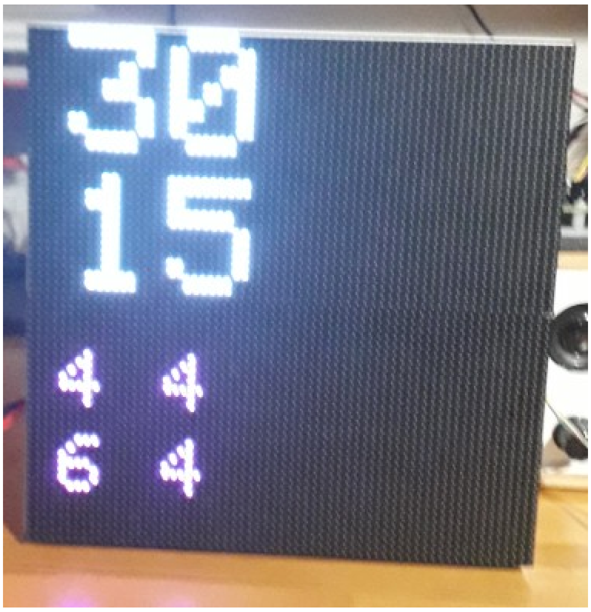
Some useful addresses

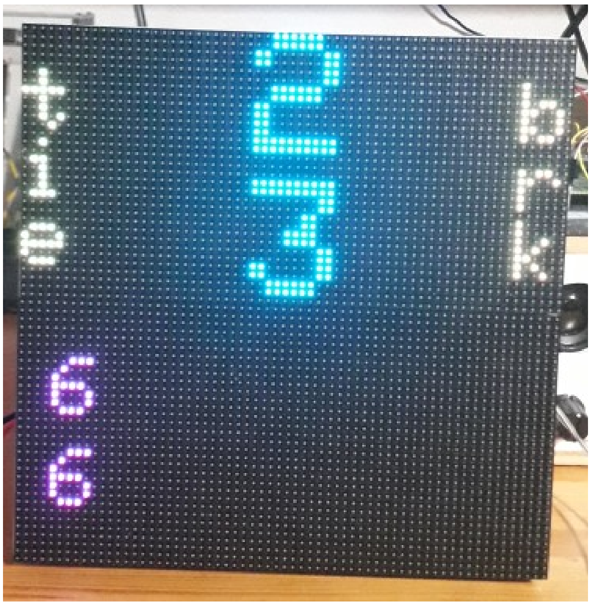
Info ESP32
https://randomnerdtutorials.com/projects-esp32/

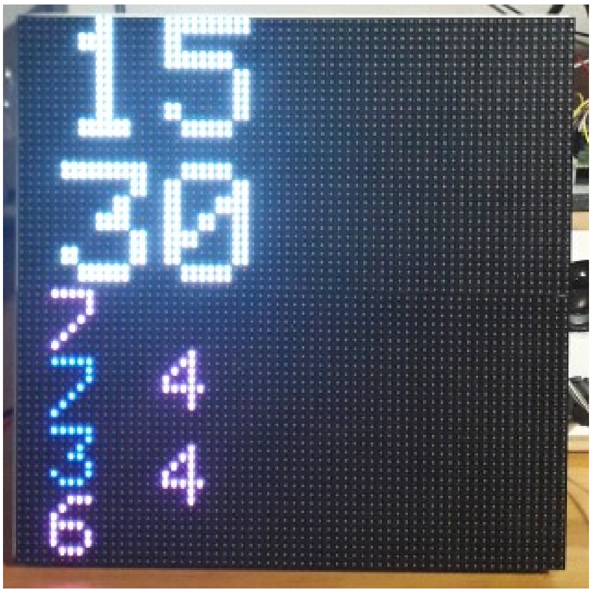
ESP32 and Arduino IDE
https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/

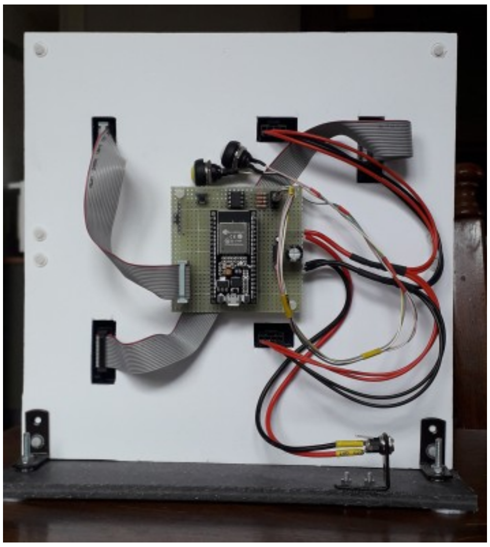
How to program
a ATTINY45
https://circuitdigest.com/microcontroller-projects/programming-attiny85-microcontroller-ic-using-arduino

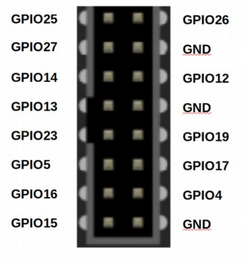
Libraries used in this program
https://github.com/peteroden/ESP32Matrix
The library included on github has already the necessary change as described later in this manual

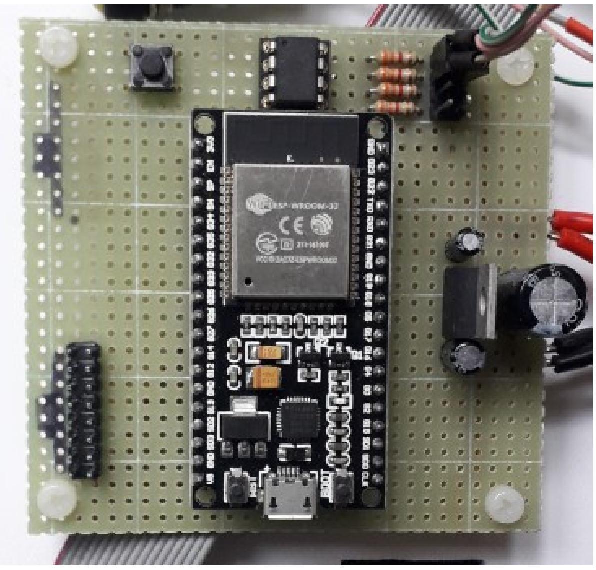

30
15
4 4
6 4

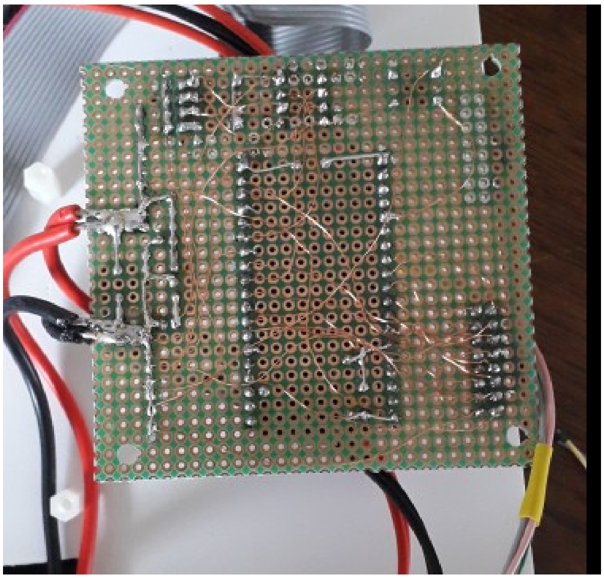

tie
23
brk
6
6


15
30
7
7 4
3 4
6




GPIO25
GPIO26
GPIO27
GND
GPIO14
GPIO12
GPIO13
GND
GPIO23
GPIO19
GPIO5
GPIO17
GPIO16
GPIO4
GPIO15
GND





Parts List

1 x ESP32 WROOM Devkit

1 x ATTINY45

2 x 64x32 LED Matrix P4

https://www.aliexpress.com/item/32754106669.html?spm=a2g0o.productlist.0.0.31a06bf9kCHEc0&algo_pvid=6f698d28-5a7d-4f4b-bfdf-4fa77f100b63&algo_exp_id=6f698d28-5a7d-4f4b-bfdf-4fa77f100b63-0&pdp_ext_f=%7B%22sku_id%22%3A%2212000028533950705%22%7D&pdp_npi=2%40dis%21EUR%2120.59%2120.59%21%21%21%21%21%402103143616621457398171239ed262%2112000028533950705%21sea&curPageLogUid=cUND3Cxf1woZ

1 x LM3940 5V to 3.3V

1 x 1000 uF 16V

1 x 100 uF 16V

1 x 4.7 uF 16V

3 x 3K3

1 x push button N.O. for resetting the score

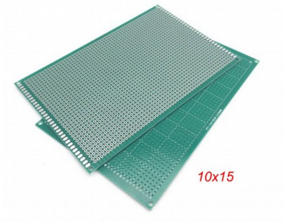
2 x push button N.O. for point counting

1 x flat cable 16 pin plug M

2 x flat cable 16 pin F/F

1 x 5 Volt / 2A

1 x PCB board single side 10x15cm



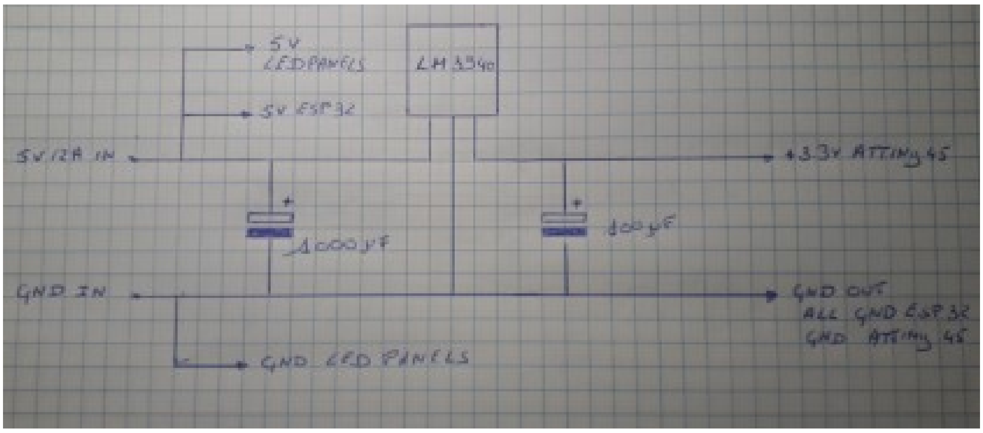
1x https://www.conrad.be/nl/p/block-koperdraad-gelakt-buitendiameter-excl-isolatielak-0-22-mm-571-m-0-20-kg-605311.html

the varnish at the begin or the end is easy to remove with the soldering tip

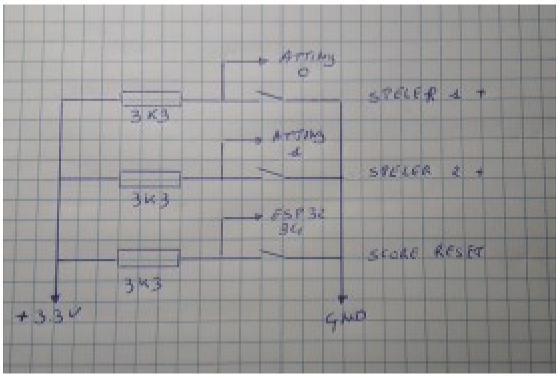
1x some soldering experience would be handy

Connections

Power



Push buttons



ESP32

Led panel (see also picture page 3)

R1	>>	GPIO25
G1	>>	GPIO26
B1	>>	GPIO27
R2	>>	GPIO14
G2	>>	GPIO12
B2	>>	GPIO13
A	>>	GPIO23
B	>>	GPIO19
C	>>	GPIO5
D	>>	GPIO17
LAT	>>	GPIO4
OE	>>	GPIO15
CLK	>>	GPIO16

ATTINY45	**>>**	**ESP32**
PB3	>>	GPIO18
PB4	>>	GPIO22
PB2	>>	GPIO21

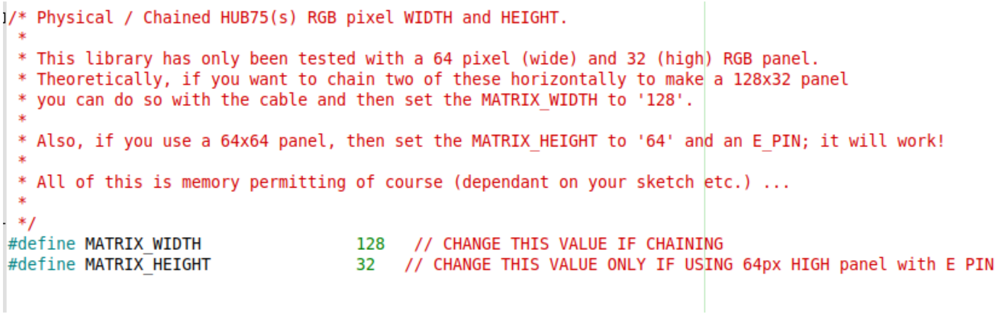
ATTINY45

PB0	>>	N.O. Player 1 +
PB1	>>	N.O. Player 2 +

ESP32

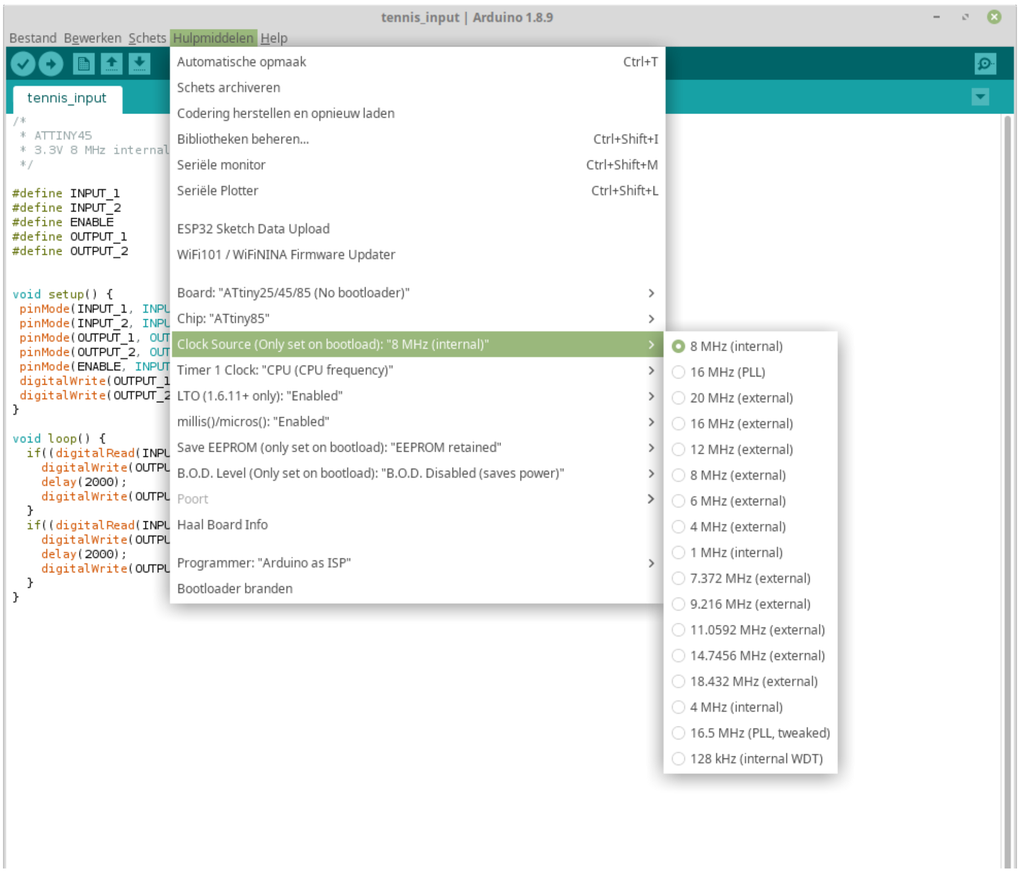
GPIO34	>>	N.O. Score Reset

After finishing soldering and connecting the flat cables as seen on the picture on page 3 it’s time to program the ESP32 Devkit
Load the program **tennis_ok_volledig_backup.ino** in the arduino IDE.
If you have installed the library from github then you can skip the following otherwise you have to change a setting in the file **ESP32-RGB64x32MatrixPanel-I2S-DMA.h** in the folder **ESP32Matrix-master** in your Arduino libraries folder .
Change #define MATRIX_WIDTH to 128 as seen on the screen print below.



```
/* Physical / Chained HUB75(s) RGB pixel WIDTH and HEIGHT.
 *
 * This library has only been tested with a 64 pixel (wide) and 32 (high) RGB panel.
 * Theoretically, if you want to chain two of these horizontally to make a 128x32 panel
 * you can do so with the cable and then set the MATRIX_WIDTH to '128'.
 *
 * Also, if you use a 64x64 panel, then set the MATRIX_HEIGHT to '64' and an E_PIN; it will work!
 *
 * All of this is memory permitting of course (dependant on your sketch etc.) ...
 *
 */
#define MATRIX_WIDTH                128   // CHANGE THIS VALUE IF CHAINING
#define MATRIX_HEIGHT               32   // CHANGE THIS VALUE ONLY IF USING 64px HIGH panel with E PIN
```

Now you can program your ESP32.
After finishing you can program the ATTINY45
Load the program **tennis_input.ino** in the Arduino IDE.
Change settings see screen print below



```
/*
 * ATTINY45
 * 3.3V 8 MHz internal
 */

#define INPUT_1
#define INPUT_2
#define ENABLE
#define OUTPUT_1
#define OUTPUT_2


void setup() {
 pinMode(INPUT_1, INPU
 pinMode(INPUT_2, INPU
 pinMode(OUTPUT_1, OUT
 pinMode(OUTPUT_2, OUT
 pinMode(ENABLE, INPUT
 digitalWrite(OUTPUT_1
 digitalWrite(OUTPUT_2
}

void loop() {
  if((digitalRead(INPU
    digitalWrite(OUTPU
    delay(2000);
    digitalWrite(OUTPU
  }
  if((digitalRead(INPU
    digitalWrite(OUTPU
    delay(2000);
    digitalWrite(OUTPU
  }
}
```

Board: "ATtiny25/45/85 (No bootloader)"
Chip: "ATtiny85"
Clock Source (Only set on bootload): "8 MHz (internal)"
Timer 1 Clock: "CPU (CPU frequency)"
LTO (1.6.11+ only): "Enabled"
millis()/micros(): "Enabled"
Save EEPROM (only set on bootload): "EEPROM retained"
B.O.D. Level (Only set on bootload): "B.O.D. Disabled (saves power)"
Programmer: "Arduino as ISP"

Follow instructions
https://circuitdigest.com/microcontroller-projects/programming-attiny85-microcontroller-ic-using-arduino
for programming the ATTINY45 (there is no difference in programming a ATTINY45 or ATTINY85).

There are a lot of other sites about how to program a ATTINY45/85 if you don’t like this one feel free to chose another.
If everything is programmed you can start playing tennis.
The score is also stored in Flash memory so you don’t lose the score by power failure.
However counting is done in an interrupt subroutine. Backup outside this subroutine if anything happens between counting and saving then there is a little problem. Best is to start with enough power in the battery (and the players).

RESET SCORE

AT *POWER-UP* PUSH AND HOLD FOR A MOMENT THE *SCORE RESET BUTTON*

Have fun
greetings
thieu

tennis_ok_volledig_backup.ino

```
/*
 * ESP32-RGB64x32MatrixPanel-I2S-DMA-master librarie
 * ESP32-RGB64x32MatrixPanel-I2S-DMA.h
 * #define MATRIX_WIDTH                128  voor 2 panelen
 * https://github.com/peteroden/ESP32Matrix
 *
 * ESP32 Dev Module
 */
#include <Preferences.h>
#include <ESP32-RGB64x32MatrixPanel-I2S-DMA.h>

Preferences pref;
RGB64x32MatrixPanel_I2S_DMA ledscherm;

#define R1_PIN  25
#define G1_PIN  26
#define B1_PIN  27
#define R2_PIN  14
#define G2_PIN  12
#define B2_PIN  13
#define A_PIN   23
#define B_PIN   19
#define C_PIN   5
#define D_PIN   17
#define E_PIN   -1
#define LAT_PIN 4
#define OE_PIN  15
#define CLK_PIN 16

#define SPELER_1_PLUS   18
#define SPELER_2_PLUS   22
#define ATTINY_ENABLE   21

#define SCORE_RESET     34

int set_nu;

int games_set_1_speler_1;
int games_set_2_speler_1;
int games_set_3_speler_1;
int games_set_4_speler_1;
int games_set_5_speler_1;

int games_set_1_speler_2;
int games_set_2_speler_2;
int games_set_3_speler_2;
int games_set_4_speler_2;
int games_set_5_speler_2;

int punten_game_speler_1;
int punten_game_speler_2;

int tiebreak_set;

int tiebreak_set_1_speler_1;
int tiebreak_set_2_speler_1;
int tiebreak_set_3_speler_1;
int tiebreak_set_4_speler_1;
int tiebreak_set_5_speler_1;

int tiebreak_set_1_speler_2;
int tiebreak_set_2_speler_2;
int tiebreak_set_3_speler_2;
int tiebreak_set_4_speler_2;
int tiebreak_set_5_speler_2;

bool ad_1;
bool ad_2;
bool tiebreak;
bool tiebreak_set_1;
bool tiebreak_set_2;
bool tiebreak_set_3;
bool tiebreak_set_4;
bool tiebreak_set_5;
bool afgewerkt = false;
bool interrupt_gezet = false;
bool gedaan;
bool data_bewaren = false;

long millis_vorig;

void beginscherm(){
  ledscherm.setTextSize(1);
  ledscherm.setTextColor(ledscherm.color333(0, 0, 255));
  ledscherm.setCursor(82, 0);
  ledscherm.println("thieu");
  ledscherm.setCursor(85, 8);
  ledscherm.println("2020");
}

void IRAM_ATTR scherm(){
  ledscherm.setTextSize(1);
  ledscherm.setTextWrap(false);
  ledscherm.setTextColor(ledscherm.color333(255,0,255));
  ledscherm.clearScreen();
  switch(set_nu){
    case 1:
      ledscherm.setCursor(4, 7);
      ledscherm.println(games_set_1_speler_1);
      ledscherm.setCursor(4, 18);
      ledscherm.println(games_set_1_speler_2);
      break;
    case 2:
      if(tiebreak_set_1 == true){
        ledscherm.setCursor(4, 0);
        ledscherm.println(games_set_1_speler_1);
        ledscherm.setTextColor(ledscherm.color333(0,0,255));
        if(tiebreak_set_1_speler_1 < 10){
          ledscherm.setCursor(4, 8);
        }
        else{
          ledscherm.setCursor(0, 8);
        }
        ledscherm.println(tiebreak_set_1_speler_1);
        if(tiebreak_set_1_speler_2 < 10){
          ledscherm.setCursor(4, 16);
        }
        else{
          ledscherm.setCursor(0, 16);
        }
        ledscherm.println(tiebreak_set_1_speler_2);
        ledscherm.setTextColor(ledscherm.color333(255,0,255));
        ledscherm.setCursor(4, 24);
        ledscherm.println(games_set_1_speler_2);
      }
      else{
        ledscherm.setCursor(4, 7);
        ledscherm.println(games_set_1_speler_1);
        ledscherm.setCursor(4, 18);
        ledscherm.println(games_set_1_speler_2);
      }
      ledscherm.setCursor(17, 7);
      ledscherm.println(games_set_2_speler_1);
      ledscherm.setCursor(17, 18);
      ledscherm.println(games_set_2_speler_2);
      break;
    case 3:
      if(tiebreak_set_1 == true){
        ledscherm.setCursor(4, 0);
        ledscherm.println(games_set_1_speler_1);
        ledscherm.setTextColor(ledscherm.color333(0,0,255));
        if(tiebreak_set_1_speler_1 < 10){
          ledscherm.setCursor(4, 8);
        }
        else{
          ledscherm.setCursor(0, 8);
        }
        ledscherm.println(tiebreak_set_1_speler_1);
        if(tiebreak_set_1_speler_2 < 10){
          ledscherm.setCursor(4, 16);
        }
        else{
          ledscherm.setCursor(0, 16);
        }
        ledscherm.println(tiebreak_set_1_speler_2);
        ledscherm.setTextColor(ledscherm.color333(255,0,255));
        ledscherm.setCursor(4, 24);
        ledscherm.println(games_set_1_speler_2);
      }
      else{
        ledscherm.setCursor(4, 7);
        ledscherm.println(games_set_1_speler_1);
        ledscherm.setCursor(4, 18);
        ledscherm.println(games_set_1_speler_2);
      }
      if(tiebreak_set_2 == true){
        ledscherm.setCursor(17, 0);
        ledscherm.println(games_set_2_speler_1);
        ledscherm.setTextColor(ledscherm.color333(0,0,255));
        if(tiebreak_set_2_speler_1 < 10){
          ledscherm.setCursor(17, 8);
        }
        else{
          ledscherm.setCursor(13, 8);
        }
        ledscherm.println(tiebreak_set_2_speler_1);
        if(tiebreak_set_2_speler_2 < 10){
          ledscherm.setCursor(17, 16);
        }
        else{
          ledscherm.setCursor(13, 16);
        }
        ledscherm.println(tiebreak_set_2_speler_2);
        ledscherm.setTextColor(ledscherm.color333(255,0,255));
        ledscherm.setCursor(17, 24);
        ledscherm.println(games_set_2_speler_2);
      }
      else{
        ledscherm.setCursor(17, 7);
        ledscherm.println(games_set_2_speler_1);
        ledscherm.setCursor(17, 18);
        ledscherm.println(games_set_2_speler_2);
      }
      ledscherm.setCursor(30, 7);
      ledscherm.println(games_set_3_speler_1);
      ledscherm.setCursor(30, 18);
      ledscherm.println(games_set_3_speler_2);
      break;
    case 4:
      if(tiebreak_set_1 == true){
        ledscherm.setCursor(4, 0);
        ledscherm.println(games_set_1_speler_1);
        ledscherm.setTextColor(ledscherm.color333(0,0,255));
        if(tiebreak_set_1_speler_1 < 10){
          ledscherm.setCursor(4, 8);
        }
        else{
          ledscherm.setCursor(0, 8);
        }
        ledscherm.println(tiebreak_set_1_speler_1);
        if(tiebreak_set_1_speler_2 < 10){
          ledscherm.setCursor(4, 16);
        }
        else{
          ledscherm.setCursor(0, 16);
        }
        ledscherm.println(tiebreak_set_1_speler_2);
        ledscherm.setTextColor(ledscherm.color333(255,0,255));
        ledscherm.setCursor(4, 24);
        ledscherm.println(games_set_1_speler_2);
      }
      else{
        ledscherm.setCursor(4, 7);
        ledscherm.println(games_set_1_speler_1);
        ledscherm.setCursor(4, 18);
        ledscherm.println(games_set_1_speler_2);
      }
      if(tiebreak_set_2 == true){
        ledscherm.setCursor(17, 0);
        ledscherm.println(games_set_2_speler_1);
        ledscherm.setTextColor(ledscherm.color333(0,0,255));
        if(tiebreak_set_2_speler_1 < 10){
          ledscherm.setCursor(17, 8);
        }
        else{
          ledscherm.setCursor(13, 8);
        }
        ledscherm.println(tiebreak_set_2_speler_1);
        if(tiebreak_set_2_speler_2 < 10){
          ledscherm.setCursor(17, 16);
        }
        else{
          ledscherm.setCursor(13, 16);
        }
        ledscherm.println(tiebreak_set_2_speler_2);;
        ledscherm.setTextColor(ledscherm.color333(255,0,255));
        ledscherm.setCursor(17, 24);
        ledscherm.println(games_set_2_speler_2);
      }
      else{
        ledscherm.setCursor(17, 7);
        ledscherm.println(games_set_2_speler_1);
        ledscherm.setCursor(17, 18);
        ledscherm.println(games_set_2_speler_2);
      }
      if(tiebreak_set_3 == true){
        ledscherm.setCursor(30, 0);
        ledscherm.println(games_set_3_speler_1);
        ledscherm.setTextColor(ledscherm.color333(0,0,255));
        if(tiebreak_set_3_speler_1 < 10){
          ledscherm.setCursor(30, 8);
        }
        else{
          ledscherm.setCursor(26, 8);
        }
        ledscherm.println(tiebreak_set_3_speler_1);
        if(tiebreak_set_3_speler_2 < 10){
          ledscherm.setCursor(30, 16);
        }
        else{
          ledscherm.setCursor(26, 16);
        }
        ledscherm.println(tiebreak_set_3_speler_2);
        ledscherm.setTextColor(ledscherm.color333(255,0,255));
        ledscherm.setCursor(30, 24);
        ledscherm.println(games_set_3_speler_2);
      }
      else{
        ledscherm.setCursor(30, 7);
        ledscherm.println(games_set_3_speler_1);
        ledscherm.setCursor(30, 18);
        ledscherm.println(games_set_3_speler_2);
      }
      ledscherm.setCursor(43, 7);
      ledscherm.println(games_set_4_speler_1);
      ledscherm.setCursor(43, 18);
      ledscherm.println(games_set_4_speler_2);
      break;
    case 5:
      if(tiebreak_set_1 == true){
        ledscherm.setCursor(4, 0);
        ledscherm.println(games_set_1_speler_1);
        ledscherm.setTextColor(ledscherm.color333(0,0,255));
        if(tiebreak_set_1_speler_1 < 10){
          ledscherm.setCursor(4, 8);
        }
        else{
          ledscherm.setCursor(0, 8);
        }
        ledscherm.println(tiebreak_set_1_speler_1);
        if(tiebreak_set_1_speler_2 < 10){
          ledscherm.setCursor(4, 16);
        }
        else{
          ledscherm.setCursor(0, 16);
        }
        ledscherm.println(tiebreak_set_1_speler_2);
        ledscherm.setTextColor(ledscherm.color333(255,0,255));
        ledscherm.setCursor(4, 24);
        ledscherm.println(games_set_1_speler_2);
      }
      else{
```

```
  ledscherm.setCursor(4, 7);
  ledscherm.println(games_set_1_speler_1);
  ledscherm.setCursor(4, 18);
  ledscherm.println(games_set_1_speler_2);
}
if(tiebreak_set_2 == true){
  ledscherm.setCursor(17, 0);
  ledscherm.println(games_set_2_speler_1);
  ledscherm.setTextColor(ledscherm.color333(0,0,255));
  if(tiebreak_set_2_speler_1 < 10){
    ledscherm.setCursor(17, 8);
  }
  else{
    ledscherm.setCursor(13, 8);
  }
  ledscherm.println(tiebreak_set_2_speler_1);
  if(tiebreak_set_2_speler_2 < 10){
    ledscherm.setCursor(17, 16);
  }
  else{
    ledscherm.setCursor(13, 16);
  }
  ledscherm.println(tiebreak_set_2_speler_2);;
  ledscherm.setTextColor(ledscherm.color333(255,0,255));
  ledscherm.setCursor(17, 24);
  ledscherm.println(games_set_2_speler_2);
}
else{
  ledscherm.setCursor(17, 7);
  ledscherm.println(games_set_2_speler_1);
  ledscherm.setCursor(17, 18);
  ledscherm.println(games_set_2_speler_2);
}
if(tiebreak_set_3 == true){
  ledscherm.setCursor(30, 0);
  ledscherm.println(games_set_3_speler_1);
  ledscherm.setTextColor(ledscherm.color333(0,0,255));
  if(tiebreak_set_3_speler_1 < 10){
    ledscherm.setCursor(30, 8);
  }
  else{
    ledscherm.setCursor(26, 8);
  }
  ledscherm.println(tiebreak_set_3_speler_1);
  if(tiebreak_set_3_speler_2 < 10){
    ledscherm.setCursor(30, 16);
  }
  else{
    ledscherm.setCursor(26, 16);
  }
  ledscherm.println(tiebreak_set_3_speler_2);
  ledscherm.setTextColor(ledscherm.color333(255,0,255));
  ledscherm.setCursor(30, 24);
  ledscherm.println(games_set_3_speler_2);
}
else{
  ledscherm.setCursor(30, 7);
  ledscherm.println(games_set_3_speler_1);
  ledscherm.setCursor(30, 18);
  ledscherm.println(games_set_3_speler_2);
}
if(tiebreak_set_4 == true){
  ledscherm.setCursor(43, 0);
  ledscherm.println(games_set_4_speler_1);
  ledscherm.setTextColor(ledscherm.color333(0,0,255));
  if(tiebreak_set_4_speler_1 < 10){
    ledscherm.setCursor(43, 8);
  }
  else{
    ledscherm.setCursor(39, 8);
  }
  ledscherm.println(tiebreak_set_4_speler_1);
  if(tiebreak_set_4_speler_2 < 10){
    ledscherm.setCursor(43, 16);
  }
  else{
    ledscherm.setCursor(39, 16);
  }
  ledscherm.println(tiebreak_set_4_speler_2);
```

```
      ledscherm.setTextColor(ledscherm.color333(255,0,255));
      ledscherm.setCursor(43, 24);
      ledscherm.println(games_set_4_speler_2);
    }
    else{
      ledscherm.setCursor(43, 7);
      ledscherm.println(games_set_4_speler_1);
      ledscherm.setCursor(43, 18);
      ledscherm.println(games_set_4_speler_2);
    }
    ledscherm.setCursor(56, 7);
    ledscherm.println(games_set_5_speler_1);
    ledscherm.setCursor(56, 18);
    ledscherm.println(games_set_5_speler_2);
    break;
  case 6:
    if(tiebreak_set_1 == true){
      ledscherm.setCursor(4, 0);
      ledscherm.println(games_set_1_speler_1);
      ledscherm.setTextColor(ledscherm.color333(0,0,255));
      if(tiebreak_set_1_speler_1 < 10){
        ledscherm.setCursor(4, 8);
      }
      else{
        ledscherm.setCursor(0, 8);
      }
      ledscherm.println(tiebreak_set_1_speler_1);
      if(tiebreak_set_1_speler_2 < 10){
        ledscherm.setCursor(4, 16);
      }
      else{
        ledscherm.setCursor(0, 16);
      }
      ledscherm.println(tiebreak_set_1_speler_2);
      ledscherm.setTextColor(ledscherm.color333(255,0,255));
      ledscherm.setCursor(4, 24);
      ledscherm.println(games_set_1_speler_2);
    }
    else{
      ledscherm.setCursor(4, 7);
      ledscherm.println(games_set_1_speler_1);
      ledscherm.setCursor(4, 18);
      ledscherm.println(games_set_1_speler_2);
    }
    if(tiebreak_set_2 == true){
      ledscherm.setCursor(17, 0);
      ledscherm.println(games_set_2_speler_1);
      ledscherm.setTextColor(ledscherm.color333(0,0,255));
      if(tiebreak_set_2_speler_1 < 10){
        ledscherm.setCursor(17, 8);
      }
      else{
        ledscherm.setCursor(13, 8);
      }
      ledscherm.println(tiebreak_set_2_speler_1);
      if(tiebreak_set_2_speler_2 < 10){
        ledscherm.setCursor(17, 16);
      }
      else{
        ledscherm.setCursor(13, 16);
      }
      ledscherm.println(tiebreak_set_2_speler_2);;
      ledscherm.setTextColor(ledscherm.color333(255,0,255));
      ledscherm.setCursor(17, 24);
      ledscherm.println(games_set_2_speler_2);
    }
    else{
      ledscherm.setCursor(17, 7);
      ledscherm.println(games_set_2_speler_1);
      ledscherm.setCursor(17, 18);
      ledscherm.println(games_set_2_speler_2);
    }
    if(tiebreak_set_3 == true){
      ledscherm.setCursor(30, 0);
      ledscherm.println(games_set_3_speler_1);
      ledscherm.setTextColor(ledscherm.color333(0,0,255));
      if(tiebreak_set_3_speler_1 < 10){
        ledscherm.setCursor(30, 8);
      }
      else{
        ledscherm.setCursor(26, 8);
      }
      ledscherm.println(tiebreak_set_3_speler_1);
      if(tiebreak_set_3_speler_2 < 10){
        ledscherm.setCursor(30, 16);
      }
      else{
        ledscherm.setCursor(26, 16);
      }
      ledscherm.println(tiebreak_set_3_speler_2);
      ledscherm.setTextColor(ledscherm.color333(255,0,255));
      ledscherm.setCursor(30, 24);
      ledscherm.println(games_set_3_speler_2);
    }
    else{
      ledscherm.setCursor(30, 7);
      ledscherm.println(games_set_3_speler_1);
      ledscherm.setCursor(30, 18);
      ledscherm.println(games_set_3_speler_2);
    }
    if(tiebreak_set_4 == true){
      ledscherm.setCursor(43, 0);
      ledscherm.println(games_set_4_speler_1);
      ledscherm.setTextColor(ledscherm.color333(0,0,255));
      if(tiebreak_set_4_speler_1 < 10){
        ledscherm.setCursor(43, 8);
      }
      else{
        ledscherm.setCursor(39, 8);
      }
      ledscherm.println(tiebreak_set_4_speler_1);
      if(tiebreak_set_4_speler_2 < 10){
        ledscherm.setCursor(43, 16);
      }
      else{
        ledscherm.setCursor(39, 16);
      }
      ledscherm.println(tiebreak_set_4_speler_2);
      ledscherm.setTextColor(ledscherm.color333(255,0,255));
      ledscherm.setCursor(43, 24);
      ledscherm.println(games_set_4_speler_2);
    }
    else{
      ledscherm.setCursor(43, 7);
      ledscherm.println(games_set_4_speler_1);
      ledscherm.setCursor(43, 18);
      ledscherm.println(games_set_4_speler_2);
    }
    ledscherm.setCursor(56, 0);
    ledscherm.println(games_set_5_speler_1);
    ledscherm.setTextColor(ledscherm.color333(0,0,255));
      if(tiebreak_set_5_speler_1 < 10){
        ledscherm.setCursor(56, 8);
      }
      else{
        ledscherm.setCursor(52, 8);
      }
      ledscherm.println(tiebreak_set_5_speler_1);
      if(tiebreak_set_5_speler_2 < 10){
        ledscherm.setCursor(56, 16);
      }
      else{
        ledscherm.setCursor(52, 16);
      }
      ledscherm.println(tiebreak_set_5_speler_2);
    ledscherm.setTextColor(ledscherm.color333(255,0,255));
    ledscherm.setCursor(56, 24);
    ledscherm.println(games_set_5_speler_2);
}
if(gedaan == true){
  ledscherm.setTextSize(1);
  ledscherm.setTextColor(ledscherm.color333(255, 0, 0));
  ledscherm.setCursor(76, 0);
  ledscherm.println("May the");
  ledscherm.setCursor(82, 8);
  ledscherm.println("Force");
  ledscherm.setCursor(76, 16);
  ledscherm.println("be with");
  ledscherm.setCursor(87, 24);
  ledscherm.println("you");
}
else{
  if(tiebreak == true){
    ledscherm.setTextSize(1);
    ledscherm.setTextColor(ledscherm.color333(255,255,0));
    ledscherm.setCursor(64, 5);
    ledscherm.print("t");
    ledscherm.setCursor(64, 13);
    ledscherm.print("i");
    ledscherm.setCursor(64, 21);
    ledscherm.print("e");
    ledscherm.setCursor(122, 5);
    ledscherm.print("b");
    ledscherm.setCursor(122, 13);
    ledscherm.print("r");
    ledscherm.setCursor(122, 21);
    ledscherm.print("k");
    ledscherm.setTextSize(2);
    ledscherm.setTextColor(ledscherm.color333(0,255,255));
    switch(tiebreak_set){
      case 1:
        if(tiebreak_set_1_speler_1 < 10){
          ledscherm.setCursor(91, 0);
        }
        else{
          ledscherm.setCursor(84, 0);
        }
        ledscherm.print(tiebreak_set_1_speler_1);
        if(tiebreak_set_1_speler_2 < 10){
          ledscherm.setCursor(91, 17);
        }
        else{
          ledscherm.setCursor(84, 17);
        }
        ledscherm.print(tiebreak_set_1_speler_2);
        break;
      case 2:
        if(tiebreak_set_2_speler_1 < 10){
          ledscherm.setCursor(91, 0);
        }
        else{
          ledscherm.setCursor(84, 0);
        }
        ledscherm.print(tiebreak_set_2_speler_1);
        if(tiebreak_set_2_speler_2 < 10){
          ledscherm.setCursor(91, 17);
        }
        else{
          ledscherm.setCursor(84, 17);
        }
        ledscherm.print(tiebreak_set_2_speler_2);
        break;
      case 3:
        if(tiebreak_set_3_speler_1 < 10){
          ledscherm.setCursor(91, 0);
        }
        else{
          ledscherm.setCursor(84, 0);
        }
        ledscherm.print(tiebreak_set_3_speler_1);
        if(tiebreak_set_3_speler_2 < 10){
          ledscherm.setCursor(91, 17);
        }
        else{
          ledscherm.setCursor(84, 17);
        }
        ledscherm.print(tiebreak_set_3_speler_2);
        break;
      case 4:
        if(tiebreak_set_4_speler_1 < 10){
          ledscherm.setCursor(91, 0);
        }
        else{
          ledscherm.setCursor(84, 0);
        }
        ledscherm.print(tiebreak_set_4_speler_1);
        if(tiebreak_set_4_speler_2 < 10){
            ledscherm.setCursor(91, 17);
          }
          else{
            ledscherm.setCursor(84, 17);
          }
          ledscherm.print(tiebreak_set_4_speler_2);
          break;
        case 5:
          if(tiebreak_set_5_speler_1 < 10){
            ledscherm.setCursor(91, 0);
          }
          else{
            ledscherm.setCursor(84, 0);
          }
          ledscherm.print(tiebreak_set_5_speler_1);
          if(tiebreak_set_5_speler_2 < 10){
            ledscherm.setCursor(91, 17);
          }
          else{
            ledscherm.setCursor(84, 17);
          }
          ledscherm.print(tiebreak_set_5_speler_2);
      }
    }
    else{
      ledscherm.setTextColor(ledscherm.color333(255,255,255));
      ledscherm.setTextSize(2);
      if(punten_game_speler_1 == 0){
        ledscherm.setCursor(81, 0);
      }
      else{
        ledscherm.setCursor(69, 0);
      }
      ledscherm.println(punten_game_speler_1);
      if(ad_1 == true){
        ledscherm.setCursor(97, 0);
        ledscherm.setTextColor(ledscherm.color333(255,255,0));
        ledscherm.println("AD");
        ledscherm.setTextColor(ledscherm.color333(255,255,255));
      }
      if(punten_game_speler_2 == 0){
        ledscherm.setCursor(81, 17);
      }
      else{
        ledscherm.setCursor(69, 17);
      }
      ledscherm.println(punten_game_speler_2);
      if(ad_2 == true){
        ledscherm.setCursor(97, 17);
        ledscherm.setTextColor(ledscherm.color333(255,255,0));
        ledscherm.println("AD");
        ledscherm.setTextColor(ledscherm.color333(255,255,255));
      }
    }
  }
}

void IRAM_ATTR plus_1(){
  digitalWrite(ATTINY_ENABLE, LOW);

  interrupt_gezet = false;
  afgewerkt = false;
  data_bewaren = true;

  millis_vorig = millis();

  if(tiebreak == true){
    switch(tiebreak_set){
      case 1:
        tiebreak_set_1_speler_1 ++;
        if((tiebreak_set_1_speler_1 > 6) && ((tiebreak_set_1_speler_1 - tiebreak_set_1_speler_2) > 1)){
          tiebreak = false;
          games_set_1_speler_1 ++;
          set_nu ++;
        }
        break;
      case 2:
        tiebreak_set_2_speler_1 ++;
        if((tiebreak_set_2_speler_1 > 6) && ((tiebreak_set_2_speler_1 - tiebreak_set_2_speler_2) > 1)){
```

```
        tiebreak = false;
        games_set_2_speler_1 ++;
        set_nu ++;
      }
      break;
    case 3:
      tiebreak_set_3_speler_1 ++;
      if((tiebreak_set_3_speler_1 > 6) && ((tiebreak_set_3_speler_1 - tiebreak_set_3_speler_2) > 1)){
        tiebreak = false;
        games_set_3_speler_1 ++;
        set_nu ++;
      }
      break;
    case 4:
      tiebreak_set_4_speler_1 ++;
      if((tiebreak_set_4_speler_1 > 6) && ((tiebreak_set_4_speler_1 - tiebreak_set_4_speler_2) > 1)){
        tiebreak = false;
        games_set_4_speler_1 ++;
        set_nu ++;
      }
      break;
    case 5:
      tiebreak_set_5_speler_1 ++;
      if((tiebreak_set_5_speler_1 > 6) && ((tiebreak_set_5_speler_1 - tiebreak_set_5_speler_2) > 1)){
        tiebreak = false;
        games_set_5_speler_1 ++;
        set_nu ++;
        gedaan = true;
      }
  }
}
else{
  if((ad_1 == true) || ((punten_game_speler_1 == 40) && (punten_game_speler_2 != 40))){
    ad_1 = false;
    punten_game_speler_1 = 0;
    punten_game_speler_2 = 0;
    switch(set_nu){
      case 1:
        games_set_1_speler_1 ++;
        if((games_set_1_speler_1 > 5) && ((games_set_1_speler_1 - games_set_1_speler_2) > 1)){
          set_nu ++;
        }
        if((games_set_1_speler_1 == 6) && (games_set_1_speler_2 == 6)){
          tiebreak_set_1 = true;
          tiebreak_set = 1;
          tiebreak = true;
        }
        break;
      case 2:
        games_set_2_speler_1 ++;
        if((games_set_2_speler_1 > 5) && ((games_set_2_speler_1 - games_set_2_speler_2) > 1)){
          set_nu ++;
        }
        if((games_set_2_speler_1 == 6) && (games_set_2_speler_2 == 6)){
          tiebreak_set_2 = true;
          tiebreak_set = 2;
          tiebreak = true;
        }
        break;
      case 3:
        games_set_3_speler_1 ++;
        if((games_set_3_speler_1 > 5) && ((games_set_3_speler_1 - games_set_3_speler_2) > 1)){
          set_nu ++;
        }
          if((games_set_3_speler_1 == 6) && (games_set_3_speler_2 == 6)){
          tiebreak_set_3 = true;
          tiebreak_set = 3;
          tiebreak = true;
        }
        break;
      case 4:
        games_set_4_speler_1 ++;
        if((games_set_4_speler_1 > 5) && ((games_set_4_speler_1 - games_set_4_speler_2) > 1)){
          set_nu ++;
        }
        if((games_set_4_speler_1 == 6) && (games_set_4_speler_2 == 6)){
          tiebreak_set_4 = true;
          tiebreak_set = 4;
          tiebreak = true;
```

```
     }
     break;
    case 5:
     games_set_5_speler_1 ++;
     if((games_set_5_speler_1 > 5) && ((games_set_5_speler_1 - games_set_5_speler_2) > 1)){
      gedaan = true;
     }
     if((games_set_5_speler_1 == 6) && (games_set_5_speler_2 == 6)){
      tiebreak_set_5 = true;
      tiebreak_set = 5;
      tiebreak = true;
     }
   }
   afgewerkt = true;
  }
  if((ad_2 == true) && (afgewerkt == false)){
   ad_2 = false;
   afgewerkt = true;
  }
  if((punten_game_speler_1 == 40) && (afgewerkt == false)){
    ad_1 = true;
    afgewerkt = true;
  }
  if((punten_game_speler_1 == 30) && (afgewerkt == false)){
   punten_game_speler_1 = 40;
   afgewerkt = true;
  }
  if((punten_game_speler_1 == 15) && (afgewerkt == false)){
   punten_game_speler_1 = 30;
   afgewerkt = true;
  }

  if((punten_game_speler_1 == 0) && (afgewerkt == false)){
   punten_game_speler_1 = 15;
   afgewerkt = true;
  }
  if(tiebreak == true){
   punten_game_speler_1 = 0;
   punten_game_speler_2 = 0;
  }
 }
 //scherm();
}

void IRAM_ATTR plus_2(){
 digitalWrite(ATTINY_ENABLE, LOW);

 interrupt_gezet = false;
 afgewerkt = false;
 data_bewaren = true;

 millis_vorig = millis();

 if(tiebreak == true){
  switch(tiebreak_set){
   case 1:
    tiebreak_set_1_speler_2 ++;
    if((tiebreak_set_1_speler_2 > 6) && ((tiebreak_set_1_speler_2 - tiebreak_set_1_speler_1) > 1)){
     tiebreak = false;
     games_set_1_speler_2 ++;
     set_nu ++;
    }
    break;
   case 2:
    tiebreak_set_2_speler_2 ++;
    if((tiebreak_set_2_speler_2 > 6) && ((tiebreak_set_2_speler_2 - tiebreak_set_2_speler_1) > 1)){
     tiebreak = false;
     games_set_2_speler_2 ++;
     set_nu ++;
    }
    break;
   case 3:
    tiebreak_set_3_speler_2 ++;
    if((tiebreak_set_3_speler_2 > 6) && ((tiebreak_set_3_speler_2 - tiebreak_set_3_speler_1) > 1)){
     tiebreak = false;
     games_set_3_speler_2 ++;
     set_nu ++;
    }
    break;
    case 4:
      tiebreak_set_4_speler_2 ++;
      if((tiebreak_set_4_speler_2 > 6) && ((tiebreak_set_4_speler_2 - tiebreak_set_4_speler_1) > 1)){
        tiebreak = false;
        games_set_4_speler_2 ++;
        set_nu ++;
      }
      break;
    case 5:
      tiebreak_set_5_speler_2 ++;
      if((tiebreak_set_5_speler_2 > 6) && ((tiebreak_set_5_speler_2 - tiebreak_set_5_speler_1) > 1)){
        tiebreak = false;
        games_set_5_speler_2 ++;
        ;
        set_nu ++;
        gedaan = true;
      }
  }
}
else{
  if((ad_2 == true) || ((punten_game_speler_2 == 40) && (punten_game_speler_1 != 40))){
    ad_2 = false;
    punten_game_speler_1 = 0;
    punten_game_speler_2 = 0;
    switch(set_nu){
      case 1:
        games_set_1_speler_2 ++;
        if((games_set_1_speler_2 > 5) && ((games_set_1_speler_2 - games_set_1_speler_1) > 1)){
          set_nu ++;
        }
        if((games_set_1_speler_2 == 6) && (games_set_1_speler_1 == 6)){
          tiebreak_set_1 = true;
          tiebreak_set = 1;
          tiebreak = true;
        }
        break;
      case 2:
        games_set_2_speler_2 ++;
        if((games_set_2_speler_2 > 5) && ((games_set_2_speler_2 - games_set_2_speler_1) > 1)){
          set_nu ++;
        }
        if((games_set_2_speler_2 == 6) && (games_set_2_speler_1 == 6)){
          tiebreak_set_2 = true;
          tiebreak_set = 2;
          tiebreak = true;
        }
        break;
      case 3:
        games_set_3_speler_2 ++;
        if((games_set_3_speler_2 > 5) && ((games_set_3_speler_2 - games_set_3_speler_1) > 1)){
          set_nu ++;
        }
        if((games_set_3_speler_2 == 6) && (games_set_3_speler_1 == 6)){
          tiebreak_set_3 = true;
          tiebreak_set = 3;
          tiebreak = true;
        }
        break;
      case 4:
        games_set_4_speler_2 ++;
        if((games_set_4_speler_2 > 5) && ((games_set_4_speler_2 - games_set_4_speler_1) > 1)){
          set_nu ++;
        }
        if((games_set_4_speler_2 == 6) && (games_set_4_speler_1 == 6)){
          tiebreak_set_4 = true;
          tiebreak_set = 4;
          tiebreak = true;
        }
        break;
      case 5:
        games_set_5_speler_2 ++;
        if((games_set_5_speler_2 > 5) && ((games_set_5_speler_2 - games_set_5_speler_1) > 1)){
          gedaan = true;
        }
        if((games_set_5_speler_2 == 6) && (games_set_5_speler_1 == 6)){
          tiebreak_set_5 = true;
          tiebreak_set = 5;
          tiebreak = true;
        }
      }
      afgewerkt = true;
    }
    if((ad_1 == true) && (afgewerkt == false)){
      ad_1 = false;
      afgewerkt = true;
    }
    if((punten_game_speler_2 == 40) && (afgewerkt == false)){
      ad_2 = true;
      afgewerkt = true;
    }
    if((punten_game_speler_2 == 30) && (afgewerkt == false)){
      punten_game_speler_2 = 40;
      afgewerkt = true;
    }
    if((punten_game_speler_2 == 15) && (afgewerkt == false)){
      punten_game_speler_2 = 30;
      afgewerkt = true;
    }
    if((punten_game_speler_2 == 0)  && (afgewerkt == false)){
      punten_game_speler_2 = 15;
      afgewerkt = true;
    }
    if(tiebreak == true){
      punten_game_speler_1 = 0;
      punten_game_speler_2 = 0;

    }
  }
  //scherm();
}

void setup() {
  ledscherm.begin(R1_PIN, G1_PIN, B1_PIN, R2_PIN, G2_PIN, B2_PIN, A_PIN, B_PIN, C_PIN, D_PIN, E_PIN, LAT_PIN, OE_PIN, CLK_PIN );

  pinMode(SPELER_1_PLUS, INPUT);
  pinMode(SPELER_2_PLUS, INPUT);
  pinMode(SCORE_RESET, INPUT);
  pinMode(ATTINY_ENABLE, OUTPUT);
  digitalWrite(ATTINY_ENABLE, LOW);
  Serial.begin(115200);

  pref.begin("score", false);

  if(digitalRead(SCORE_RESET) == false){
    pref.putInt("s_n", 1);
    pref.putInt("g_s_1_s_1", 0);
    pref.putInt("g_s_2_s_1", 0);
    pref.putInt("g_s_3_s_1", 0);
    pref.putInt("g_s_4_s_1", 0);
    pref.putInt("g_s_5_s_1", 0);

    pref.putInt("g_s_1_s_2", 0);
    pref.putInt("g_s_2_s_2", 0);
    pref.putInt("g_s_3_s_2", 0);
    pref.putInt("g_s_4_s_2", 0);
    pref.putInt("g_s_5_s_2", 0);

    pref.putInt("p_g_s_1", 0);
    pref.putInt("p_g_s_2", 0);

    pref.putInt("t_s", 0);

    pref.putInt("t_s_1_s_1", 0);
    pref.putInt("t_s_2_s_1", 0);
    pref.putInt("t_s_3_s_1", 0);
    pref.putInt("t_s_4_s_1", 0);
    pref.putInt("t_s_5_s_1", 0);

    pref.putInt("t_s_1_s_2", 0);
    pref.putInt("t_s_2_s_2", 0);
    pref.putInt("t_s_3_s_2", 0);
    pref.putInt("t_s_4_s_2", 0);
    pref.putInt("t_s_5_s_2", 0);

    pref.putBool("a_1", false);
    pref.putBool("a_2", false);

    pref.putBool("pref.tiebreak", false);
    pref.putBool("t_s_1", false);
    pref.putBool("t_s_2", false);
    pref.putBool("t_s_3", false);
    pref.putBool("t_s_4", false);
    pref.putBool("t_s_5", false);
    pref.putBool("g", false);
  }

  set_nu = pref.getInt("s_n");

  games_set_1_speler_1 = pref.getInt("g_s_1_s_1");
  games_set_2_speler_1 = pref.getInt("g_s_2_s_1");
  games_set_3_speler_1 = pref.getInt("g_s_3_s_1");
  games_set_4_speler_1 = pref.getInt("g_s_4_s_1");
  games_set_5_speler_1 = pref.getInt("g_s_5_s_1");

  games_set_1_speler_2 = pref.getInt("g_s_1_s_2");
  games_set_2_speler_2 = pref.getInt("g_s_2_s_2");
  games_set_3_speler_2 = pref.getInt("g_s_3_s_2");
  games_set_4_speler_2 = pref.getInt("g_s_4_s_2");
  games_set_5_speler_2 = pref.getInt("g_s_5_s_2");

  punten_game_speler_1 = pref.getInt("p_g_s_1");
  punten_game_speler_2 = pref.getInt("p_g_s_2");

  tiebreak_set = pref.getInt("t_s");

  tiebreak_set_1_speler_1 = pref.getInt("t_s_1_s_1");
  tiebreak_set_2_speler_1 = pref.getInt("t_s_2_s_1");
  tiebreak_set_3_speler_1 = pref.getInt("t_s_3_s_1");
  tiebreak_set_4_speler_1 = pref.getInt("t_s_4_s_1");
  tiebreak_set_5_speler_1 = pref.getInt("t_s_5_s_1");

  tiebreak_set_1_speler_2 = pref.getInt("t_s_1_s_2");
  tiebreak_set_2_speler_2 = pref.getInt("t_s_2_s_2");
  tiebreak_set_3_speler_2 = pref.getInt("t_s_3_s_2");
  tiebreak_set_4_speler_2 = pref.getInt("t_s_4_s_2");
  tiebreak_set_5_speler_2 = pref.getInt("t_s_5_s_2");

  ad_1 = pref.getBool("a_1");
  ad_2 = pref.getBool("a_2");

  tiebreak  = pref.getBool("t");
  tiebreak_set_1 = pref.getBool("t_s_1");
  tiebreak_set_2 = pref.getBool("t_s_2");
  tiebreak_set_3 = pref.getBool("t_s_3");
  tiebreak_set_4 = pref.getBool("t_s_4");
  tiebreak_set_5 = pref.getBool("t_s_5");
  gedaan = pref.getBool("g");

  beginscherm();

  millis_vorig = millis();
  while((millis() - millis_vorig) < 2500){
  }

  scherm();

  millis_vorig = millis();

  attachInterrupt(SPELER_1_PLUS, plus_1, FALLING);
  attachInterrupt(SPELER_2_PLUS, plus_2, FALLING);
}

void loop() {
  if(data_bewaren == true){
    data_bewaren = false;
    pref.putInt("t_s_1_s_1", tiebreak_set_1_speler_1);
    pref.putInt("t_s_2_s_1", tiebreak_set_2_speler_1);
    pref.putInt("t_s_3_s_1", tiebreak_set_3_speler_1);
    pref.putInt("t_s_4_s_1", tiebreak_set_4_speler_1);
    pref.putInt("t_s_5_s_1", tiebreak_set_5_speler_1);
    pref.putBool("t_s_1", tiebreak_set_1);
    pref.putBool("t_s_2", tiebreak_set_2);
    pref.putBool("t_s_3", tiebreak_set_3);
    pref.putBool("t_s_4", tiebreak_set_4);
    pref.putBool("t_s_5", tiebreak_set_5);
    pref.putInt("g_s_1_s_1", games_set_1_speler_1);
    pref.putInt("g_s_2_s_1", games_set_2_speler_1);
    pref.putInt("g_s_3_s_1", games_set_3_speler_1);
    pref.putInt("g_s_4_s_1", games_set_4_speler_1);
    pref.putInt("g_s_5_s_1", games_set_5_speler_1);
    pref.putInt("p_g_s_1", punten_game_speler_1);
    pref.putInt("p_g_s_2", punten_game_speler_2);
    pref.putBool("t", tiebreak);
    pref.putInt("s_n", set_nu);
    pref.putBool("g", gedaan);
    pref.putBool("a_1", ad_1);
    pref.putBool("a_2", ad_2);
    pref.putInt("t_s", tiebreak_set);
    pref.putInt("t_s_1_s_2", tiebreak_set_1_speler_2);
    pref.putInt("t_s_2_s_2", tiebreak_set_2_speler_2);
    pref.putInt("t_s_3_s_2", tiebreak_set_3_speler_2);
    pref.putInt("t_s_4_s_2", tiebreak_set_4_speler_2);
    pref.putInt("t_s_5_s_2", tiebreak_set_5_speler_2);
    pref.putInt("g_s_1_s_2", games_set_1_speler_2);
    pref.putInt("g_s_2_s_2", games_set_2_speler_2);
    pref.putInt("g_s_3_s_2", games_set_3_speler_2);
    pref.putInt("g_s_4_s_2", games_set_4_speler_2);
    pref.putInt("g_s_5_s_2", games_set_5_speler_2);
    scherm();
  }
  if(((millis() - millis_vorig) > 2500) && (interrupt_gezet == false)){
    digitalWrite(ATTINY_ENABLE, HIGH);
    interrupt_gezet = true;
  }
}
```

tennis_input.ino

```
/*
MIT License

Copyright (c) 2022 thieu-b55

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
*/
/*
 * ATTINY45
 * 3.3V 8 MHz internal
 */

#define INPUT_1       0
#define INPUT_2       1
#define ENABLE        2
#define OUTPUT_1      3
#define OUTPUT_2      4


void setup() {
 pinMode(INPUT_1, INPUT);
 pinMode(INPUT_2, INPUT);
 pinMode(OUTPUT_1, OUTPUT);
 pinMode(OUTPUT_2, OUTPUT);
 pinMode(ENABLE, INPUT);
 digitalWrite(OUTPUT_1, true);
 digitalWrite(OUTPUT_2, true);
}

void loop() {
  if((digitalRead(INPUT_1) == false) && (digitalRead(ENABLE) == true)){
    digitalWrite(OUTPUT_1, false);
    delay(2000);
    digitalWrite(OUTPUT_1, true);
  }
  if((digitalRead(INPUT_2) == false) && (digitalRead(ENABLE) == true)){
    digitalWrite(OUTPUT_2, false);
    delay(2000);
    digitalWrite(OUTPUT_2, true);
  }
}
```