

Software voor de livestream lcd en liquidCrystal op zij van meterkast Mariakerk

livestream.ino

```
#include <SPI.h> //Load the SPI Library
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
/* omdat arduino niet internet en I2C tegelijk kan, daarom is deze arduino uno toegevoegd
 * hij krijgt de signalen liveaan -- muziekaan binnen op D4-D7
 * ze komen van de andere arduino die ze uitgeeft op D1-D3 let op dat je de gnd tussen beide
 arduino's niet vergeet!!
 */
const int liveaan = 4;
const int volgaan = 5;
const int micaan = 6;
const int muziekaan = 7;
int liveaanstatus = 0;
int volgaanstatus = 0;
int micaanstatus = 0;
int muziekaanstatus = 0;
int volgaanvorig = 2;
int micaanvorig = 2;
int muziekaanvorig = 2;
int nietactief = 2;
int line = 0;
int allesok;
int klaar = 0;
int toggle = 0;
LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16 chars and 2 line
display

void setup() {
  Serial.begin(9600); //Turn on Serial Port
  delay(1500); //delay
  pinMode(liveaan, INPUT);
  pinMode(volgaan, INPUT);
  pinMode(micaan, INPUT);
  pinMode(muziekaan, INPUT);
  lcd.init(); // initialize the lcd
  lcd.backlight();
  allesok = 0;
}

void loop()
{
  liveaanstatus = digitalRead(liveaan);
  Serial.println("");
  Serial.println("Begonnen.");
  Serial.print("liveaanstatus");
```

```

Serial.print(liveaanstatus);
Serial.println("");
Serial.print("volgaanstatus");
Serial.print(volgaanstatus);
Serial.println("");
Serial.print("micaanstatus");
Serial.print(micaanstatus);
Serial.println("");
Serial.print("muziekaanstatus");
Serial.print(muziekaanstatus);
klaar = 0;
if(liveaanstatus == 1) // anders geen zien
{
  nietactief = 1;
  volgaanstatus = digitalRead(volgaan);
  micaanstatus = digitalRead(micaan);
  muziekaanstatus = digitalRead(muziekaan);
  if(volgaanstatus == volgaanvorig)
  {
    klaar++;
  }
  if(micaanstatus == micaanvorig)
  {
    klaar++;
  }
  if(muziekaanstatus == muziekaanvorig)
  {
    klaar++;
  }
}
if(allesok !=3)
{
  if(toggle > 2) // om de 2/3 passen dus
  {
    clearLCDLine(0);
    lcd.setCursor(1,0);
    lcd.print("Livestream actief!");
    toggle = -1;
  }
  else
  {
    if(toggle == 0)
    {
      clearLCDLine(0);
      lcd.setCursor(0,0);
      lcd.print("Let op instellingen!");
    }
    toggle++;
  }
}
}

```

```

if(klaar == 3)
{
  delay(1000);
  volgaanvorig = volgaanstatus;
  micaanvorig = micaanstatus;
  muziekaanvorig = muziekaanstatus;
  return;
}
allesok = 0;
clearLCDLine(1);
clearLCDLine(2);
clearLCDLine(3);
if(volgaanstatus == 0)
{
  lcd.setCursor(0,1);
  lcd.print("Volgmij = UIT");
}
else
{
  allesok++;
}
if(micaanstatus == 1)
{
  lcd.setCursor(0,2);
  lcd.print("Micro's = UIT");
}
else
{
  allesok++;
}
if(muziekaanstatus == 1)
{
  lcd.setCursor(0,3);
  lcd.print("Muziek = AAN (ext.)");
  lcd.setCursor(0,2);
}
else
{
  allesok++;
}
if (allesok == 3)
{
  clearLCDLine(0);
  lcd.setCursor(0,0);
  lcd.print("-----");
  clearLCDLine(1);
  lcd.setCursor(0,1);
  lcd.print("| Livestream aan |");
  clearLCDLine(2);
}

```

```

    lcd.setCursor(0,2);
    lcd.print("| ALLES IN ORDE! |");
    clearLCDLine(3);
    lcd.setCursor(0,3);
    lcd.print("-----");
    toggle = 4;
}
}
else
{
    if(nietactief != 0)
    {
        allesok = 0;
        clearLCDLine(0);
        clearLCDLine(1);
        clearLCDLine(2);
        clearLCDLine(3);
        lcd.setCursor(2,1);
        lcd.print("GEEN LIVESTREAM");
        lcd.setCursor(7,2);
        lcd.print("ACTIEF");
        lcd.setCursor(0,3);
        nietactief = 0;
    }
    liveaanstatus = 0;
    volgaanstatus = 0;
    micaanstatus = 0;
    muziekaanstatus = 0;
    klaar = 0;
}
volgaanvorig = volgaanstatus;
miceaanvorig = micaanstatus;
muziekaanvorig = muziekaanstatus;
delay(1000);
}

void clearLCDLine(int line)
{
    for(int n = 0; n < 20; n++) { // 20 indicates symbols in line. For 2x16 LCD write - 16
        lcd.setCursor(n,line);
        lcd.print(" ");
    }
}
}

```