

## Software om de microfoontoetsen naar OBS te sturen, als toetsaanslag.

### volgmij\_exta\_17.ino

```
#include <SPI.h>
#include <Ethernet.h>
#include <Keyboard.h>
// dit is de software van het volgmij-systeem
// stuurt OBS via hotkeys en OBS stuurt via 192.168.2.177 door naar tally
// bij elke verandering van aan/uit gebeurt dit, maar ook elke 10 sec. bv. als deze arduino
tussentijds uitgeschakeld werd
// dan krijgt tally anders bij inschakelen geen update van de stand van zaken van de aan/uit
knopDA
// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = { 0xDE, 0xAA, 0xBE, 0xEF, 0xFE, 0xED };
// if you don't want to use DNS (and reduce your sketch size)
// use the numeric IP instead of the name for the server:
IPAddress server(192,168,2,177); // arduino die de tally bedient, hoort uiteindelijk 177 te zijn
// Set the static IP address to use if the DHCP fails to assign
IPAddress ip(192, 168, 2, 178);
IPAddress myDns(192, 168, 2, 1);
EthernetClient client;
// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to (port 80 is default for HTTP):
const int volgaan = 0; // moet toggle worden hoog ... volgaanled aan
const int beameraan = 1;
const int geelaan = 2; // dit is beeld-aan vanuit projectie daar aansluiting 4
const int dienstbegint = 3;
const int titeltoonbaar = 4;
const int ambo = 5;
const int altaar = 6;
const int koor = 7;
const int reserve1 = 8;
const int reserve2 = 9;
const int volgaanled = A0;
// let op internet shield gebruikt D10-13
int beameraanstatus = 0; // is dus uit
int geelaanstatus = 0; // current status of the button beeld is uit.
int lastgeelaanstatus = 0; // uit ..
int altaarstatus = 0; // le op alls neg!! 0 is dus altaarstatus is aan, ombouwen naar positief
ev.
int lastaltaarstatus = 0;
int ambostatus = 0;
int lastambostatus = 0;
int koorstatus = 0;
int lastkoorstatus = 0;
int reserve1status = 0;
```

```

int lastreserve1status = 0;
int reserve2status    = 0;
int lastreserve2status = 0;
int volgaanstatus     = 0;
volatile bool volgaantoggle = true; //in aan starten
int titeltoonbaarstatus = 0; ///////////////
int lastvolgaanstatus = 2;
int randomnummer      = 0;
int x                  = 0;
unsigned long vorigemillis = 0;
unsigned long vorigemillis2 = 0;
unsigned long vorigemillis3 = 0;
char vorigetoets      = '8'; // willekeurig, mag dus niet gebruikt
int automaat         = 0; // deze wordt 1 bij dienstbegint

void setup()
{
  Keyboard.begin();

  delay(2000);
  pinMode(beameraan, INPUT);
  pinMode(geelaan, INPUT);
  pinMode(ambo, INPUT);
  pinMode(altaar, INPUT);
  pinMode(koor, INPUT);
  pinMode(reserve1, INPUT);
  pinMode(reserve2, INPUT);
  pinMode(volgaan, INPUT);
  pinMode(dienstbegint, INPUT);
  pinMode(titeltoonbaar, INPUT);
  pinMode(volgaanled, OUTPUT);
  Serial.begin(9600);
  interrupts ();
  delay(1000);
  attachInterrupt(digitalPinToInterrupt(volgaan), volgaangaataan, RISING);
  attachInterrupt(digitalPinToInterrupt(dienstbegint), dienstbegintaan, RISING); // als
titeltoonbaar 1 wordt en deze hoog gaat -> dienst begint => volgaanstatus = 1/ ook microfoon
en muziek moeten hier bediend worden
  delay(1000);
  Serial.println("Initialize Ethernet with DHCP:");
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // Check for Ethernet hardware present
    if (Ethernet.hardwareStatus() == EthernetNoHardware)
    {
      Serial.println("Ethernet shield was not found. Sorry, can't run without hardware. :(");
      while (true)
      {
        delay(1); // do nothing, no point running without Ethernet hardware
      }
    }
  }
}

```

```

    }
  }
  if (Ethernet.linkStatus() == LinkOFF)
  {
    Serial.println("Ethernet cable is not connected.");
  }
  // try to configure using IP address instead of DHCP:
  Ethernet.begin(mac, ip, myDns);
}
else
{
  Serial.print(" DHCP assigned IP ");
  Serial.println(Ethernet.localIP());
}
// give the Ethernet shield a second to initialize:
delay(1000);
Serial.print("connecting to ");
Serial.print(server);
Serial.println("...");
digitalWrite(volgaanled, true); // in aan starten
}

void loop()
{
  beameraanstatus = digitalRead(beameraan);
  altaarstatus    = digitalRead(altaar);
  ambostatus      = digitalRead(ambo);
  koorstatus      = digitalRead(koor);
  reserve1status  = digitalRead(reserve1);
  reserve2status  = digitalRead(reserve2);
  titeltoonbaarstatus = digitalRead(titeltoonbaar);
  volgaanstatus   = digitalRead(volgaanled); // let op essentieel hier
  randomnummer    = random(1, 10);
  x                = randomnummer % 2;
  // let op x enkel bij beameraanstatus=1 random!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  // anders is er slechts 1 keuze
  /*
  Serial.print("beameraanstatus ");
  Serial.print(beameraanstatus);
  Serial.println();
  Serial.print("geelaanstatus ");
  Serial.print(geelaanstatus);
  Serial.println();
  Serial.print("altaarstatus ");
  Serial.print(altaarstatus);
  Serial.println();
  Serial.print("ambostatus ");
  Serial.print(ambostatus);
  Serial.println();
  */
}

```

```

Serial.print("koorstatus ");
Serial.print(koorstatus);
Serial.println();
Serial.print("titeltoonbaarstatus ");
Serial.print(titeltoonbaarstatus);
Serial.println();
Serial.print("lastkoorstatus ");
Serial.print(lastkoorstatus);
Serial.println();
Serial.print("volgaanstatus "); // aanvullen
Serial.print(volgaanstatus);
Serial.println();
*/

// hier insmokkelen dat bij alle microfoons uit, het beeld centraal gaat bij volgen van
microfoons na 10 sec

// bij volgaanstatus gebeurt er eigenlijk niets, de camera staat waar deze handmatig is
ingesteld
if(volgaanstatus == 1)
{
    if ((altaarstatus == 0) && (ambostatus == 0) && (koorstatus == 0) && (reserve1status
== 0) && (reserve2status == 0))
    {
        // dus volgen staat aan en alle microfoons uit, dan beeld centraal op altaar
        if(millis() - vorigemillis > 10000) // om de 10 sec. moet 10000 zijn opnieuw zenden
vw. mogelijk aan/uit zetten op volgmij
        {
            vorigemillis = millis();
            if(titeltoonbaarstatus == 0) // geen titel tonen dan kaal altaar tonen
            {
                ctrltoets('S'); // Scene
                // Serial.println("Ctrl-S: Scene");
            }
            else
            {
                ctrltoets('Z');
                // Serial.println("Ctrl-Z: met titel"); // aanvullen
            }
        }
    }
}
if(millis() - vorigemillis2 > 5000) // om de 5 sec doen vw. mogelijk aan/uit zetten op
volgmij
{
    vorigemillis2 = millis();
    volgaanstatus == digitalRead(volgaanled);
    verstuur();
}

```

```

geelaanstatus = digitalRead(geelaan);
if ((geelaanstatus != lastgeelaanstatus) && (beameraanstatus == 1)) // dus beamerstatus aan
{
  lastgeelaanstatus = geelaanstatus;
  if (geelaanstatus == 0) // dus uit
  {
    ctrltoets('C');
    // Serial.println("Ctrl-C geel uit"); // toon beeld zonder NDlink
  }
  else
  {
    ctrltoets('D');
    // Serial.println("Ctrl-D geel aan"); // toon beeld van computer sacristie met NDlink
  }
}
if (volgaanstatus == 1) // het deel dat beamerbeeld aan/uit volgt is dus los hiervan
{
  altaarstatus = digitalRead(altaar);
  ambostatus = digitalRead(ambo);
  koorstatus = digitalRead(koor);
  reserve1status = digitalRead(reserve1);
  reserve2status = digitalRead(reserve2);
  if ((altaarstatus != lastaltaarstatus) && (altaarstatus == 1))
  {
    if(beameraanstatus == 1)
    {
      if (x == 1)
      {
        // Serial.println("Ctrl-L altaar preset 21 beamer aan");
        ctrltoets('L');
      }
      else
      {
        ctrltoets('T');
        // Serial.println("Ctrl-T altaar preset 31 beamer aan");
      }
    }
    else
    {
      ctrltoets('A');
      // Serial.println("Ctrl-A altaar preset 41 beamer uit");
    }
  }
}
if ((ambostatus != lastambostatus) && (ambostatus == 1))
{
  if(beameraanstatus == 1)
  {
    if (x == 1)

```

```

    {
        ctrltoets('M');
    //  Serial.println("Ctrl-M ambo preset 22 beamer aan");
    }
    else
    {
        ctrltoets('U');
    //  Serial.println("Ctrl-U ambo preset 32 beamer aan");
    }
}
else
{
    ctrltoets('B');
    //  Serial.println("Ctrl-B ambo preset 42 beamer uit");
}
}
if ((koorstatus != lastkoorstatus) && (koorstatus == 1))
{
    if (beameraanstatus == 1)
    {
        if (x == 1)
        {
            ctrltoets('K');
            // Serial.println("Ctrl-K koor preset 23 beamer aan");
        }
        else
        {
            ctrltoets('V');
            //Serial.println("Ctrl-V koor preset 33 beamer aan");
        }
    }
    else
    {
        ctrltoets('F');
        //Serial.println("Ctrl-F koor preset 43 beamer uit");
    }
}
// bij reserve is 2de beeld en beeld bij beameraan niet uitgewerkt
/*
if ((reserve1status <> lastreserve1status) && (reserve1status == 1))
{
    lastreserve1status = reserve1status;
    ctrltoets('R');
}
if ((reserve2status <> lastreserve2status) && (reserve2status == 1))
{
    lastreserve2status = reserve2status;
    ctrltoets('S');
}
}

```

```

    */
}
lastaltaarstatus = altaarstatus;
lastambostatus = ambostatus;
lastkoorstatus = koorstatus;
lastreserve1status = reserve1status;
lastreserve2status = reserve2status;
if (automaat == 1)
{
    vorigetoets = '8'; // willekeurig, om te voorkomen dat volgende niet worden uitgevoerd
    ctrltoets('3'); // ext. muziek uit
    ctrltoets('2'); // micro aan plaatje
    ctrltoets('6'); // audioinvoer installatie = mic aan 5 is overigens
    verstuur();//////////////////////////////////// deze hier toegevoegd 27/2/24
    automaat = 0; // anders moet je eerst resetten voor volgende dienst
    // micro aan is ctrl-2 en ctrl-6 nl. 2 functies: microf.plaatje weghalen en unmute
    // micro uit ter info is ctrl-1 en ctrl-5
}
delay(500);
}

void ctrltoets(char toets)
{
    if(toets != vorigetoets)
    {
        // enkel voor test
        Serial.print("toets CTRL-");
        Serial.print(toets);
        Serial.println(); // enkel voor test
        vorigetoets = toets;
        Keyboard.press(KEY_RIGHT_CTRL);
        Keyboard.press(toets);
        delay(50);
        Keyboard.releaseAll();
        delay(10);
    }
}

void verstuur()
{
    // if you get a connection, report back via serial:
    if (client.connect(server,80))
    {
        Serial.println("connected to");
        Serial.println(client.remoteIP());
        if(volgaanstatus == 0)
        {
            client.println("/?volguit");
            Serial.println("/?volguit");
        }
    }
}

```

```

    digitalWrite(volgaanled, LOW);
}
if(volgaanstatus == 1)
{
    if(automaat == 0)
    {
        client.println("/?volgaan");
        Serial.println("/?volgaan");
        digitalWrite(volgaanled, HIGH);
    }
    else
    {
        client.println("/?automaat");
        Serial.println("/?automaat");
        digitalWrite(volgaanled, HIGH);
    }
}
client.println("Connection: close");
client.println();
}
else
{
    Serial.println("connection failed"); // bv. als andere arduino niet aanstaat
}
if (!client.connected())
{
    // if the server is disconnected, stop the client:
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();
}
}

void volgaangaataan()
{
    if(millis() - vorigemillis3 > 50)
    {
        vorigemillis3 = millis();
        if(digitalRead(volgaan) == 1)
        {
            volgaantoggle = !volgaantoggle;
            digitalWrite(volgaanled, volgaantoggle);
        }
    }
}

void dienstbegintaan()
{
    if(digitalRead(dienstbegint) == 1)

```



```
{  
  // micro en externe muziek worden niet via volgmij gestuurd, enkel alles normaal bij  
dienstbegin  
  volgaanstatus = 1;  
  automaat      = 1;  
  volgaantoggle = true;  
  digitalWrite(volgaanled, volgaantoggle);  
}  
}
```