****

**Bluetooth communication**

1. inleiding
2. HC-05 Serial Bluetooth Module
3. AT Commands
4. Program button led
5. Program servo potmeter
6. Program oled
7. Program interrupt

Overview:

Met een afstands bediening kunnen we met de robot communiceren, de robot luistert dan als het ware naar wat wij zeggen en kunnen we vertellen wat hij moet uitvoeren.

Dit houdt wel in dat hij moet begrijpen wat wij vertellen protocol.

En in staat zijn dat uit te voeren wat wij vragen.

Dit betekend dat de robot moet kunnen luisteren maar daarnaast ook een taak moet uitvoeren.

Dit wordt al wat lastiger, omdat de robot maar 1 ding tegelijk kan.

Dit wordt over het algemeen opgelost door het af te wisselen.

Er komt een commando binnen bepaal prioriteit zet in de wacht handel af waar je mee bezig was

En voer daarna het commando uit.

Wacht tot ik groen licht geef vertel het me dan nog eens.

Wij zien dat je bezig bent

Herhaal commando tot we antwoord krijgen

Overrule commando

Benodigd heden:

Hardware:

HC-05 2x

Arduino nano

Weerstand 1 KΩ

Weerstand 2 KΩ

Button

led

Draad

Breadboard

Usb kabel

Pc

Software:

Arduino IDE

library

Wij gebruiken de HC-05

Op de markt zijn twee types HC-05 waarbij de ene 5 volt tolerant is en een voltage divider heeft ingebouwd. De ander heeft dit niet en zal er voor gezorgt moeten worden dat er 3.3 Volt op de Rx

line komt.

We hebben voor deze een voltage divider nodig, omdat de arduino nano 5 volt afgeeft.

We zullen deze in een kabel implementeren, zodat we hier geen omkijken meer naar hebben.

1K

Rx

D3

2K

Tx

D2

GND

Vcc

**Aansluiten**

Afbeelding met tekst, diagram, stroomkring, schermopname

Automatisch gegenereerde beschrijving

**Instellen HC-05 als master met behulp van arduino.**

Zodra we de HC-05 op de juist manier hebben aangesloten, kunnen we deze programmeren.

Dit kunnen we doen met behulp van een arduino, mits deze minimaal 2 hardware poorten heeft.

Voor een nano gebruiken we softwareserial kunnen we dit programma gebruiken.

Eerst zullen we moeten zorgen dat de twee HC-05 ’s elkaar vinden en met elkaar kunnen communiceren.

Met AT commando’s kunnen we een HC-05 als slave instellen en de ander als master.

Download eerst dit programma naar de arduino nano.

#include <SoftwareSerial.h>

SoftwareSerial Bluetooth(3, 2);

void setup() {

  Serial.begin(9600);

  Serial.println("HC-05 AT Commands");

  Serial.println("Function                | Command              | Response             | Parameter");

  Serial.println("Test                    | AT                   | OK                   | None");

  Serial.println("Name                    | AT+NAME?             | +NAME:HC-05, OK      | HC-05 (Default)");

  Serial.println("Password                | AT+PSWD?             | +PSWD:1234, OK       | 1234 (Default)");

  Serial.println("Change Name             | AT+NAME = <name>     |  OK                  | None");

  Serial.println("Change Password         | AT+PSWD = <pswd>     |  OK                  | None");

  Serial.println("UART                    | AT+UART?             |  +UART:38400,0,0     | 38400 (Default");

  Serial.println("Connection mode         | AT+CMODE?            |  +CMOD:<cmod>        | 0 (Default");

  Serial.println("Connection mode set     | AT+CMODE=<cmod>      |  OK                  | (0 (Connect to a specified address)) >> (1 (Connect to any available address");

  Serial.println("Find Address            | AT+ADDR?             |  +ADDR:<addr>, OK    | None");

  Serial.println("Find Address            | AT+BIND=<addr>       |  OK                  | Replace the : with ,");

  Serial.println("Reset Device            | AT+RESET             |  OK                  | None");

  Serial.println("Restore Factory Default | AT+ORGL              |  OK                  | None");

  Serial.println("");

  Serial.println("Enter AT commands to start:");

  Bluetooth.begin(38400);  //HC-05 default speed in AT command mode

}

void loop() {

  if (Bluetooth.available())

   Serial.write(Bluetooth.read());

  if (Serial.available())

    Bluetooth.write(Serial.read());

}

Om de HC-05 als slave in te stellen geven we de volgende commando’s in de arduino monitor.

AT onvangen OK terug

AT+

AT+ADDR ontvangen adres bewaar dit, aangezien we het later nodig hebben.

Vervang de HC-05 door de andere om deze als master in te stellen.

AT ontvang OK

AT

AT+BIND= address

Let op het adres ingeven kan zeer lastig zijn:

Dit wordt

Voor het adres 0:13:145B3 moet je de indeling als volgt opsplitsen:

* **NAP**: 0000 (omdat het ontbrekende deel wordt aangevuld met nullen)
* **UAP**: 13
* **LAP**: 145B3 (eveneens aangevuld met nullen om op zes tekens uit te komen, dus 0145B3)

AT+BIND=0000,13,0145B3

13:28:05.127 -> Enter AT commands to start:

13:28:39.929 -> +PIN:"1234"

13:28:39.929 -> OK

13:29:18.638 -> +NAME:H-C-2010-06-01

MAC Address 00:00:13:01:45:CF

De HC-05’s kunnen nu met elkaar verbinden.

**Programmeren HC-05**

**Button led**

Event transmitter

Button click

Method

sent message

Bluetooth receive

Bluetooth

transmit

Method

receive message

Led on

Button click

Method

transmit message

Bluetooth receive

Bluetooth

transmit

Method

receive message

Led on

We kunnen nu een arduino nano programmeren met commando’s om naar de andere arduino te sturen.

Hiervoor sluiten we een button en een led aan bij beide arduino’s sender en receiver. klikken we op de button bij de sender dan zal de led bij de receiver aan gaan.

De code kunnen we naar beide arduino’s downloaden.

#include <SoftwareSerial.h>

// Bluetooth module is connected to pins 3 (TX) and 2 (RX)

SoftwareSerial Bluetoothserial(3, 2);

const int Button = 4; // Button connected to pin 4

const int led = 5;    // LED connected to pin 5

void setup() {

    Serial.begin(9600);                // Start Serial Monitor

    Bluetoothserial.begin(38400);      // Start Bluetooth communication with HC-05

    pinMode(Button, INPUT\_PULLUP);     // Configure button as input with pull-up

    pinMode(led, OUTPUT);              // Configure LED as output

    digitalWrite(led, LOW);            // Ensure LED is initially off

    Serial.println("Ready for Bluetooth communication");

}

void loop() {

    // Check if the button is pressed

    if (digitalRead(Button) == LOW) {

        delay(50); // Debounce delay

        if (digitalRead(Button) == LOW) { // Double-check button state

            Bluetoothserial.write('b'); // Send character 'b' over Bluetooth

            Serial.println("Button pressed: 'b' sent to other board");

            delay(300); // Prevent multiple rapid sends

        }

    }

    // Check if there is incoming data from Bluetooth

    if (Bluetoothserial.available() > 0) {

        // Lees alle beschikbare data in één keer om bufferproblemen te voorkomen

        char receivedChar = Bluetoothserial.read();

        // Debugging: Toon ASCII waarde van ontvangen karakter

        Serial.print("Received from Bluetooth (ASCII): ");

        Serial.println((int)receivedChar);

        Serial.print("Received from Bluetooth (Char): ");

        Serial.println(receivedChar);

        // Turn on LED if the correct character ('b') is received

        if (receivedChar == 'b') {

            digitalWrite(led, HIGH);

            Serial.println("LED turned ON");

            // Houd de LED 2 seconden aan voordat deze uitgaat

            delay(2000);

            digitalWrite(led, LOW);

            Serial.println("LED turned OFF");

        }

    }

}

**Oled display**

De sender kunnen we ook voorzien van een oled display

Waar we berichten of acties op kunnen tonen die we verzenden en ontvangen.

Oled display

Aansluiten ic2

Library gebruiken

Programmeren

Een Event roept de bijbehorende functie op. De functie voert uit wat de functie beschrijft.

Verstuur via bluetooth een bericht

Toon op het oled display de bijbehorende tekst

Ontvangen we een bericht

Toon het bericht

Probleem is dat het tijd kost om een bericht te ontvangen

Te versturen op het display te tonen controleren of een event optreedt.

Afbeelding met diagram, Rechthoek, Plan, ontwerp

Automatisch gegenereerde beschrijving

button

Potmeter

Servo

**Master Code:**

*/\**

*\* How to configure and pair two HC-05 Bluetooth Modules*

*\* by Dejan Nedelkovski, www.HowToMechatronics.com*

*\**

*\* == MASTER CODE ==*

*\*/*

**#define ledPin 9**

int state = 0;

int potValue = 0;

void **setup**() {

pinMode(ledPin, OUTPUT);

digitalWrite(ledPin, LOW);

Serial.begin(38400); *// Default communication rate of the Bluetooth module*

}

void **loop**() {

if(Serial.available() > 0){ *// Checks whether data is comming from the serial port*

state = Serial.read(); *// Reads the data from the serial port*

}

*// Controlling the LED*

if (state == '1') {

digitalWrite(ledPin, HIGH); *// LED ON*

state = 0;

}

else if (state == '0') {

digitalWrite(ledPin, LOW); *// LED ON*

state = 0;

}

*// Reading the potentiometer*

potValue = analogRead(A0);

int potValueMapped = map(potValue, 0, 1023, 0, 255);

Serial.write(potValueMapped); *// Sends potValue to servo motor*

delay(10);

}Code language: Arduino (arduino)

**Slave Code:**

*/\**

*\* How to configure and pair two HC-05 Bluetooth Modules*

*\* by Dejan Nedelkovski, www.HowToMechatronics.com*

*\**

*\* == SLAVE CODE ==*

*\*/*

**#include <Servo.h>**

**#define button 8**

Servo myServo;

int state = 20;

int buttonState = 0;

void **setup**() {

pinMode(button, INPUT);

myServo.attach(9);

Serial.begin(38400); *// Default communication rate of the Bluetooth module*

}

void **loop**() {

if(Serial.available() > 0){ *// Checks whether data is comming from the serial port*

state = Serial.read(); *// Reads the data from the serial port*

}

*// Controlling the servo motor*

myServo.write(state);

delay(10);

*// Reading the button*

buttonState = digitalRead(button);

if (buttonState == HIGH) {

Serial.write('1'); *// Sends '1' to the master to turn on LED*

}

else {

Serial.write('0');

}

}

We kunnen nu de sender commando’s laten sturen naar de receiver om

Sequences af te spelen. Op het oled display kunnen we zien welk commando we verstuurd hebben.

Serial

Hoe werkt serial

Rx tx gebruiken hardware serial

Software serial

Lipo lader

**Bluetooth met android app**

HC-05 ontvanger Android zender

De HC-05 hadden we als zender deze kunnen we ook vervangen door een android telefoon.

Hiervoor zetten we een app op de android die het verstuurt naar de HC-05 ontvanger.

Mit app inventor

Android studio

Het onderhavige voorstel acht ik onacceptabel. Het feit dat SABIC de advocaatkosten op de schadevergoeding in mindering wenst te brengen, legt de last van deze kosten op mijn schouders, terwijl SABIC hiermee afwijkt van de eerder gemaakte afspraken. Dit lijkt een herhaling van hun handelswijze inzake de transitievergoeding, waarbij eerder toezeggingen niet werden nagekomen.

Ik heb reeds afgezien van een volledige schadevergoeding en verzocht om een bedrag dat enkel mijn directe noden dekt. Het is dan ook teleurstellend en zorgwekkend dat SABIC wederom tracht haar aansprakelijkheid te beperken, zonder enige blijk van empathie of rechtvaardigheid jegens mij als getroffen partij.

Daarbij wordt SABIC ook al geholpen hun schade al aanzienlijk beperkt wordt door het uwv.

Dit wekt de indruk dat SABIC uitsluitend haar eigen belangen nastreeft, ten koste van de belangen en het welzijn van de gedupeerde. Indien SABIC niet bereid blijkt om tot een rechtvaardige en redelijke regeling te komen, behoud ik mij het recht voor om dit geschil onder de aandacht te brengen van de moedermaatschappij SABIC en Aramco. Ik zal mijn zorgen rechtstreeks aan de directie kenbaar maken. Dit de reputatie van Sabic enorm zal schaden.

Ondanks een dat door de inspectie duidelijkk naar voren is gebracht wil sabic hier niet aan geloven en verder gaan met hun onveilige aanpak.

Leggen ze elk advies hierover naas zich neer en geen verbetering met het zal dan ook niet verwonderend zijn als een fabriek de lucht in vliegt.

Ook te zien aan het personeel wat overspannen thuis zit

Alah

Behalve dat ze hun slachtoffers slecht behandelen zien ze ook niet in dat ze op de verkeerde manier bezig zijn met veilig heid gezien het onderzoek leggen ze dit naast zich neer ipv dit op te pakken om onder de boete uit te komen vinden ze het belangrijker hoger beroep.

Het is dan ook niet verbasend als een fabriek de lucht in vliegt.

Verschillende leiding gevenden zijn zitten al overspannen thuis om dat ze niet met de gevaarlijke situaties om kunnen gaan.

Proberen ze de schade af te dingen alsof je een tweede hands auto koopt

Akkoord ga met het bedrag 180,000 euro

Daarnaast zal ik sabic schrijven

-arabieren op de hoogte brengen van

- mijn zegje zal doen tijdens het hoger beroep

Ze zullen het niet leuk vinden.

Foto in de media met dank aan sabic wat ik zal opblazen.

Nano expansion board

Nano every

Nano ble

Nano connect

Raspberry pi pico

Afbeelding met tekst, diagram, Plan, schematisch

Automatisch gegenereerde beschrijving

Expansie bord aansluiting voor

voltage regulator

bluetooth

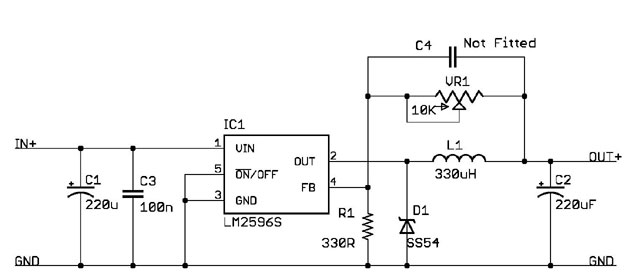
esp 32

arduino nano

arduino nano every

arduino nano ble

arduino nano rp2040



Commando walk ontvangen voer dan uit:

trot()

void trot()

Stap\_01()

Servo\_1 speed position

Servo\_2 speed position

Servo\_3 speed position

Stap\_02()

Servo\_1 speed position

Servo\_2 speed position

Servo\_3 speed position

Stap\_03

Servo\_1 speed position

Servo\_2 speed position

Servo\_3 speed position

Stap\_04

Servo\_1 speed position

Servo\_2 speed position

Servo\_3 speed position

Stap\_05

Servo\_1 speed position

Servo\_2 speed position

Servo\_3 speed position

Stap\_06

Servo\_1 speed position

Servo\_2 speed position

Servo\_3 speed position

delay()

Stap\_07

Servo\_1 speed position

Servo\_2 speed position

Servo\_3 speed position

Stap\_08

Servo\_1 speed position

Servo\_2 speed position

Servo\_3 speed position

}

Stap(speed1, position1, speed2, position2, speed3, position3)

Servo\_1 speed position

Servo\_2 speed position

Servo\_3 speed position

delay(15)

Nou moet je ook zelf leren programmeren.

Dit betekend niet gewoon beginnen met wat code in te typen en kijken wat het brengt.

Net zoals een huis gebouwd wordt begint dat met een plan of beter gezegt met jou wensen hoe het huis er uit moet zien wat het moet hebben etc.

Allemaal zaken waar je vooraf over moet nadenken en beslissingen moet nemen.

Nano servo bluetooth nano buttons oled joystick bluetooth

Nano servo esp32 xbox controller

Nano rp2040 servo xbox / nano bluetooth

Nano pololu maestro servo bluetooth

**Bluetooth xbox controller**

Esp32 C3 mini

Arduino nano

Connection

**Program esp32 C3 mini**

#include <Wire.h>

#include <Bluepad32.h>

#define I2C\_SLAVE\_ADDRESS 8

ControllerPtr myControllers[BP32\_MAX\_GAMEPADS];

void onConnectedController(ControllerPtr ctl) {

    for (int i = 0; i < BP32\_MAX\_GAMEPADS; i++) {

        if (myControllers[i] == nullptr) {

            myControllers[i] = ctl;

            Serial.printf("Controller connected, index=%d\n", i);

            break;

        }

    }

}

void onDisconnectedController(ControllerPtr ctl) {

    for (int i = 0; i < BP32\_MAX\_GAMEPADS; i++) {

        if (myControllers[i] == ctl) {

            myControllers[i] = nullptr;

            Serial.printf("Controller disconnected, index=%d\n", i);

            break;

        }

    }

}

void sendControllerData(ControllerPtr ctl) {

    if (!ctl || !ctl->isConnected())

        return;

    uint16\_t buttons = ctl->buttons();

    uint8\_t dpad = ctl->dpad();

    uint8\_t axisX = map(ctl->axisX(), -512, 511, 0, 255);

    uint8\_t axisY = map(ctl->axisY(), -512, 511, 0, 255);

    // Debug: Druk gegevens af

    Serial.printf("Sending data: Buttons=0x%04X, D-pad=0x%02X, AxisX=%d, AxisY=%d\n", buttons, dpad, axisX, axisY);

    // Verzend gegevens via I2C

    Wire.beginTransmission(I2C\_SLAVE\_ADDRESS);

    Wire.write((buttons >> 8) & 0xFF); // MSB van knoppen

    Wire.write(buttons & 0xFF);        // LSB van knoppen

    Wire.write(dpad);                  // D-pad

    Wire.write(axisX);                 // X-as

    Wire.write(axisY);                 // Y-as

    Wire.endTransmission();

}

void processControllers() {

    for (auto ctl : myControllers) {

        if (ctl && ctl->isConnected() && ctl->hasData()) {

            sendControllerData(ctl);

        }

    }

}

void setup() {

    Serial.begin(115200);

    Wire.begin(); // I2C master initialiseren

    BP32.setup(&onConnectedController, &onDisconnectedController);

    BP32.forgetBluetoothKeys(); // Reset Bluetooth-pairing indien nodig

    Serial.println("ESP32 Bluepad32 I2C Master gestart.");

}

void loop() {

    BP32.update(); // Update controllerstatussen

    processControllers();

    delay(100); // Vertraag om dataoverdracht te reguleren

}

**Arduino nano code**

#include <Wire.h>

void receiveEvent(int bytes) {

    Serial.print("Bytes received: ");

    Serial.println(bytes);

    if (bytes < 5) {

        Serial.println("Incomplete data received");

        return;

    }

    uint16\_t buttons = (Wire.read() << 8) | Wire.read(); // Knoppen

    uint8\_t dpad = Wire.read();                          // D-pad

    uint8\_t axisX = Wire.read();                         // X-as

    uint8\_t axisY = Wire.read();                         // Y-as

    // Data tonen

    Serial.print("Received Data -> Buttons: 0x");

    Serial.print(buttons, HEX); // Hexadecimal output

    Serial.print(", D-pad: 0x");

    Serial.print(dpad, HEX);    // Hexadecimal output

    Serial.print(", AxisX: ");

    Serial.print(axisX);        // Decimal output

    Serial.print(", AxisY: ");

    Serial.println(axisY);      // Decimal output

}

void setup() {

    Wire.begin(8); // Stel Arduino Nano in als I2C slave met adres 8

    Wire.onReceive(receiveEvent);

    Serial.begin(38400); // Start de serial monitor

    Serial.println("Arduino Nano I2C Slave gestart.");

}

void loop() {

    delay(100);

}

Nu kunnen we de code aanpassen :

De buttons van de xbox controller

A

B

Y

X

Joystick 1 left thumbstick

Joystick 2 right thumbstick

Directional pad D-pad

Left

Right

Up

down

view button

menu button

share button

Right trigger

Left trigger

Left bumper

Right bumper

We kunnen de voorbeeld code uploaden naar de esp32

En in de serial monitor kijken wat elke button doet

Dit zullen we moeten noteren om erachter te komen welke button wat geeft.

Als we de A button in drukken en loslaten

Hebben we eenmaal de juiste code bij de bijbehorende button

Kunnen we daar mee verder

Op de esp32 ontvangen we de button presses deze zullen we via I2C door sturen naar de arduino nano, die dan op zijn beurt een functie oproept en uitvoert.

XBOX Controller

PWM

bluetooth

Esp32

I2C

Arduino nano

servo

