

QUADROCOPTER

Door Simon Schits en Steven De Vlieger

3PBA EA-ICT

Quadrocopter

PBA elektronica-ict

door Simon Schits en Steven De Vlieger

Inleiding:

De quadrocopter bestaat uit: 4 motoren, 4 motorsturingen, een arduino nano, een draadloze ontvanger en een carbon frame. Elke motor wordt afzonderlijk aangestuurd via I²C door zijn eigen motorsturing. De arduino nano is het “brein” van de quadrocopter.

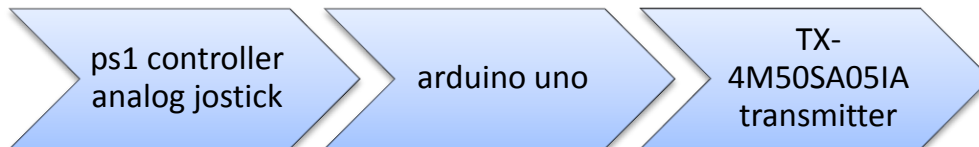
Het zendgedeelte bestaat uit een oude playstation controller, een arduino uno en een draadloze zender. De arduino uno voert een A/D conversie uit op de uitgangen van de joysticks van de playstation controller. Aan de hand van de positie van deze joysticks zullen de motoren van de quadrocopter worden aangestuurd. De data word in een array van 4bytes (1 byte voor elke motor) verzonden naar de quadrocopter.

1. eenvoudig blokschema (hardware)

1.1.ontvanger



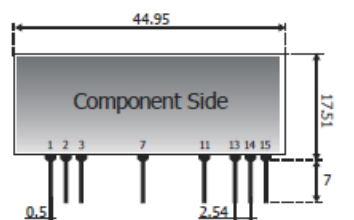
1.2. zender



2. componenten en modules

2.1.ontvanger

Dit is een draadloze ontvanger RX module type RX-4M50SA60SF Receiver van het merk Aurel.

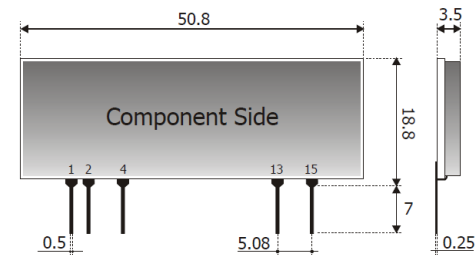


Pin 2-7-11	Ground	GND Connections: Internally connected to a single ground plate
Pin 3	Antenna	50Ω inpedance antenna connection
Pin 1-15	+V	Connection to the +5V ±0.25V supply positive pole
Pin 13	Test Point	Analog output of the demodulated signal. By connecting an oscillograph the entity and the quality of the received signal can be seen.
Pin 14	Data Out.	Receiver digital output. Aplly loads over 10 KΩ

2.2. zender

dit is de draadloze zender TX-4M50SA05IA van het merk Aurel.

CHARACTERISTICS		MIN	TYP	MAX	UNIT
V _s	Supply Voltage • Alimentazione	4.5	5	5.5	Vdc
I _s	Supply Current • Corrente assorbita		26		mA±2
F _c	Carrier frequency • Frequenza portante		433.92		MHz±75KHz
P _{RF}	RF Output power (E.R.P.) • Potenza di uscita RF (E.R.P.)	5	6	7	dBm
F _M	Square wave modulation • Frequenza di modulazione			3	KHz
L _I	Input logic level • Livello logico d'ingresso	0		V _s	V
T _{OP}	Operating temperature range • Temperatura di lavoro	-20		+80	°C



2.3. Arduino Nano

Wij gebruiken voor dit project de arduino om de quadcopter aan te sturen.
Deze is dus verbonden met de ontvanger.

Let op de pinout van deze arduino zijn gewisseld bij vorige versies

Interrupt	COM	PWM	Arduino	AVR pin	AVR pin	Arduino	Other	COM
	RXD		D0	PD0		VIN		
	TXD		D1	PD1		GND		
			Reset	PC8		Reset		
			GND	GND		5V		
INT0			D2	PD2	ADC7	A7		
INT1		Timer2B	D3	PD3	ADC6	A6		
			D4	PD4	PC5 (ADC5)	A5		SCL
		Timer0B	D5	PD5	PC4 (ADC4)	A4		SDA
		Timer0A	D6	PD6	PC3 (ADC3)	A3		
			D7	PD7	PC2 (ADC2)	A2		
			D8	PB0	PC1 (ADC1)	A1		
		Timer1A	D9	PB1	PC0 (ADC0)	A0		
	SS	Timer1B	D10	PB2	AREF	AREF		
	MOSI	Timer2A	D11	PB3	3V3			
	MISO		D12	PB4	PB5	D13	LED	SCK

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

2.4. Arduino uno

We gebruiken de arduino uno als microcontroller om de zender aan de sturen.
Deze is dus verbonden met de zender.

Specificaties:

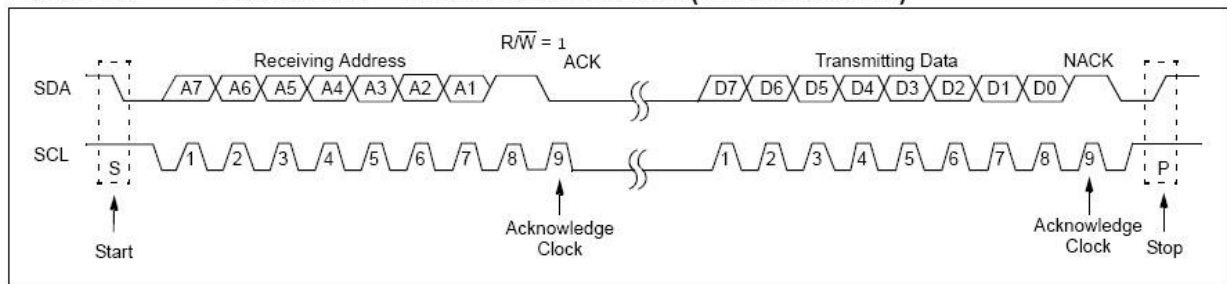
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz



2.5. I2C protocol

De arduino nano maakt verbinding met de motor control units via het I2C protocol.

FIGURE 2: TYPICAL I²C™ READ TRANSMISSION (7-BIT ADDRESS)



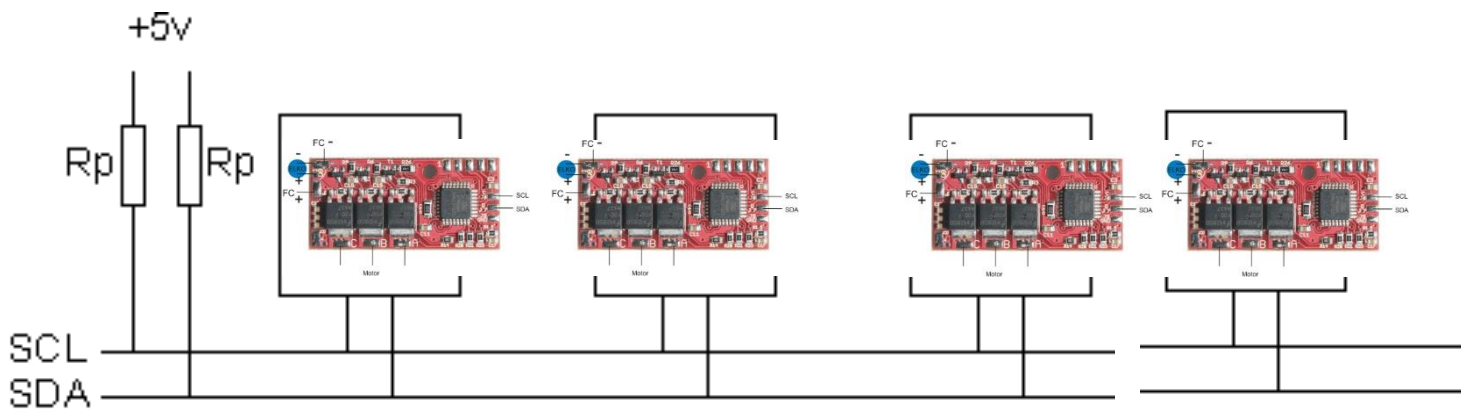
Er zijn 4 motorsturingen, 1 voor elke motor. Deze zijn in parallel verbonden met de SCL lijn en de SDA lijn.

Het protocol om I2C te gebruiken in arduino is vereenvoudigd door gebruik te maken van de Wire.h bibliotheek. `#include <Wire.h>`

Elk apparaat dat aangesloten is op de I2C bus heeft een adres.

Voor de motorsturingen zijn dit: 0X52, 0X52, 0X54 en 0X55

Belangrijk is ook om 2 weerstanden van 4,7k Ohm, een aan de SDA en een aan de SCL te verbinden met de 5V pin van de arduino (PULL-up)

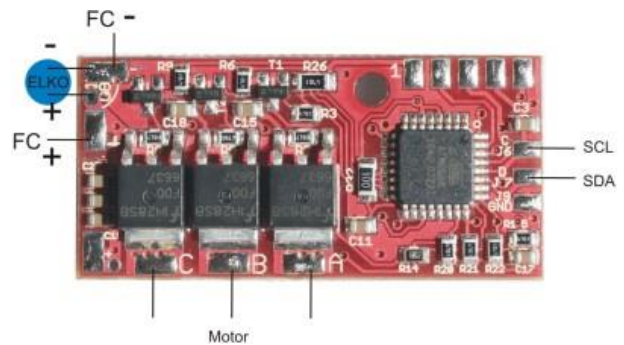
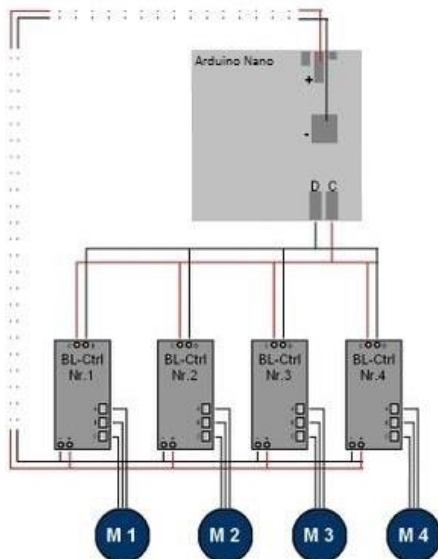


2.6. BL_Ctrl motorsturing

We gebruiken 4 motorsturingen van het type BL_Ctrl V1.1

Deze regelen de snelheid van de motoren aan de hand van de data die ze ontvangen van de arduino-nano.

- FC- is verbonden met de grond.
- FC+ is verbonden met de VCC(12V).
- De 3 motor aansluitingen gaan naar de motor
- en de SCL en SDA zijn aangesloten aan de i2C bus.



2.7. motoren

Dit zijn 4 Roxxy BL Outrunner 2827/34 motoren

- Werkspanning: 7-12V
- Max stroom: 10A
- Belastings stroom: 9A

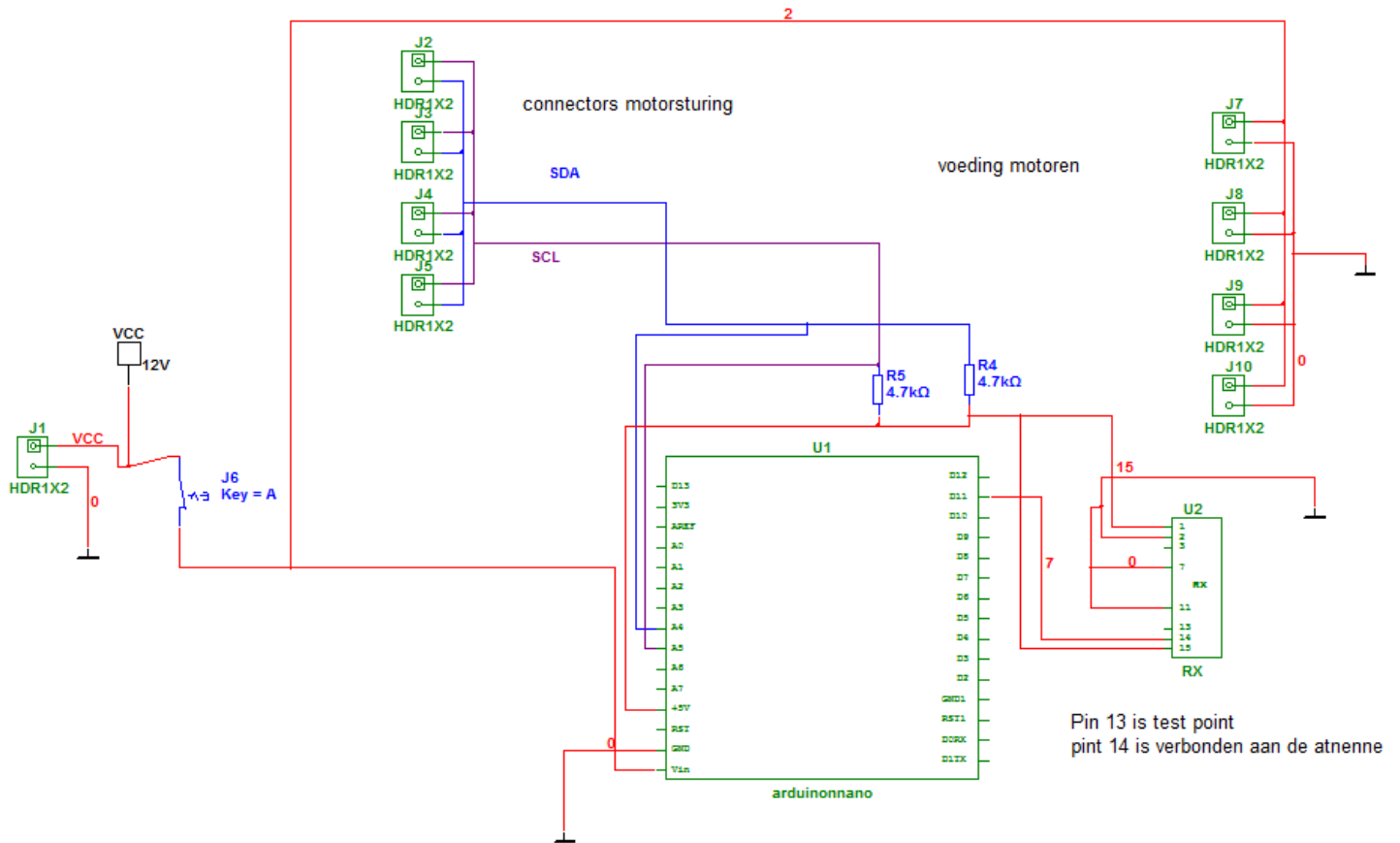


3. Aansluitingen

3.1. Schema (ontvanger)

Hier ziet u het schema van de ontvanger.

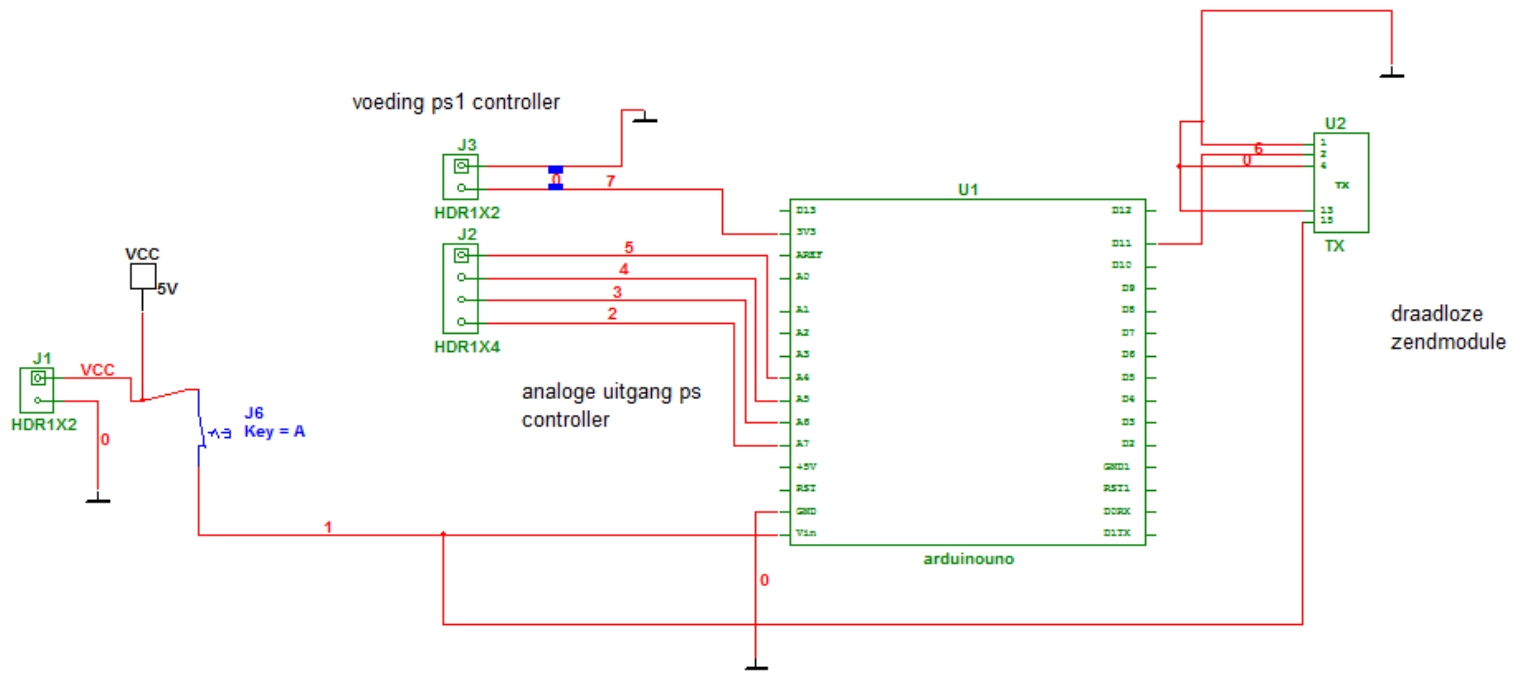
De aansluiting van de antenne zijn niet verbonden om het schema.



3.2. Schema (zender)

Hier ziet u het schema van de zender.

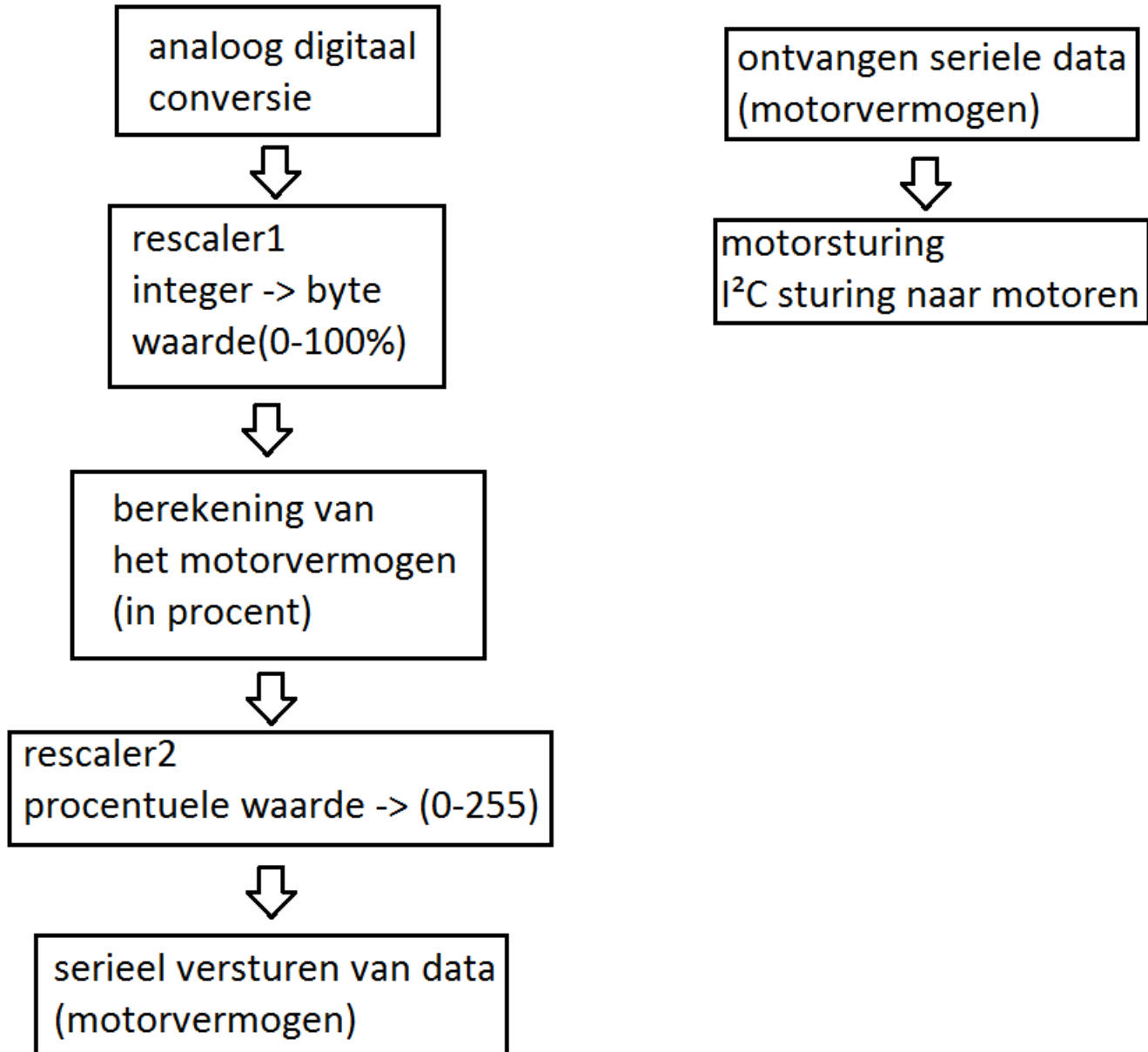
schema verbinding arduino nano met ps1 controller en zendmodule



4.1 eenvoudig blokschema (software)

Zender

ontvanger:



Arduino programma:

Zender:

```
zender_quad  functions
#include <VirtualWire.h>

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  vw_setup(2000);
}

int leftpoty;
int rightpotx;
int rightpoty;
unsigned char byteleftpoty;
unsigned char byterightpotx;
unsigned char byterightpoty;
char motor[4];
signed char rekenvar;
signed char motorvar[4];

void loop() {

  leftpoty = analogRead(A0);
  rightpotx = analogRead(A1);
  rightpoty = analogRead(A2);

  byteleftpoty = rescaler1(leftpoty);
  byterightpotx = rescaler1(rightpotx);
  byterightpoty = rescaler1(rightpoty);

  for(int i=0;i<4;i++){
    motorvar[i] = 100 - byteleftpoty;
  }

  if (byterightpotx > 50){
    rekenvar = byterightpotx - 50;
    motorvar[0] = motorvar[0] - rekenvar;
    motorvar[3] = motorvar[3] - rekenvar;
  }

  if (byterightpotx < 50){
    rekenvar = 50 - byterightpotx;
    motorvar[1] = motorvar[1] - rekenvar;
    motorvar[2] = motorvar[2] - rekenvar;
  }

  if (byterightpoty > 50){
    rekenvar = byterightpoty - 50;
    motorvar[2] = motorvar[2] - rekenvar;
    motorvar[3] = motorvar[3] - rekenvar;
  }

  if (byterightpoty < 50){
    rekenvar = 50 - byterightpoty;
    motorvar[0] = motorvar[0] - rekenvar;
    motorvar[1] = motorvar[1] - rekenvar;
  }

  for(int i=0;i<4;i++){
    motorvar[i] = positivecheck(motorvar[i]);
  }

  for(int i=0;i<4;i++){
    motor[i] = rescaler2(motorvar[i]);
  }

  vw_send((uint8_t *)motor, strlen(motor));

  Serial.println(motor[0]);
  Serial.println(motor[1]);
  Serial.println(motor[2]);
  Serial.println(motor[3]);

}
```

```
zender_quad  functions
unsigned char rescaler1(float potposition){
  float rekenvar;
  unsigned char returnval;

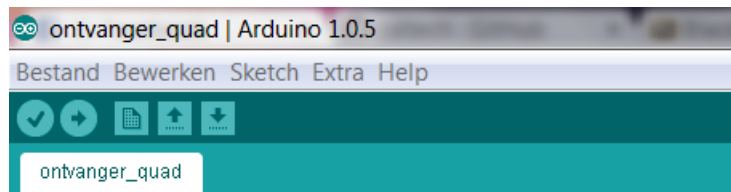
  rekenvar = potposition / 675;
  returnval = rekenvar * 100;
  return returnval;
}

unsigned char rescaler2(float motorpower){
  float rekenvar;
  unsigned char returnval;

  rekenvar = motorpower / 100;
  returnval = rekenvar * 255;
  return returnval;
}

signed char positivecheck(signed char motorpowervar){
  if (motorpowervar < 0){
    motorpowervar = 0;
  }
  return motorpowervar;
}
```

Ontvanger:



```
ontvanger_quad | Arduino 1.0.5
Bestand Bewerken Sketch Extra Help

ontvanger_quad

#include <Wire.h>
#include <VirtualWire.h>

int Sensor1Data;

char speed;
char motor = 0;
char motorspeed[4];

void setup() {
  #define BL_CTRL 0X52
  delay(1000);
  Wire.begin();
  Serial.begin(9600);
  vw_set_ptt_inverted(true);
  // Bits per sec
  vw_setup(2000);
  // Start the receiver PLL running
  vw_rx_start();
}

void motorsturing(char motor, char speed){

  Wire.beginTransaction((BL_CTRL + ( motor << 1) >>1));
  Wire.write(speed);
  Wire.endTransmission();

  delay(10);
}

void loop(){
  uint8_t buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;

  if (vw_get_message(buf, &buflen)){

    for (int i = 0; i < buflen; i++){
      motorspeed[i] = char(buf[i]);
    }
  }
}
```

Handige weetjes over de quadrocopter:

- Het frame van de quadrocopter is geleidend! Opgepast voor kortsluiting!
- De motorsturingen zijn afgeplakt met tape om contact met het frame te voorkomen.
de draden die hieruit komen afgeplakt met tape zijn de I²C lijnen (zwart = clock, rood = data)
,de andere paren draden zijn stroomdraden.
- de quadrocopter werkt op het een spanning tussen 7V - 12V en voert een stroom van ongeveer 6A bij vol vermogen.
- De motoren hebben elk een andere draairichting. Bij het installeren van de rotors moet hier rekening mee gehouden worden.
- De motorsturing wordt aangestuurd via I²C. vergeet hierbij de pull-up weerstanden niet!