Verslagen Labo mobcom

Opdracht 1 : knipperlicht

Code:

// included libary

#include <Printers.h>

#include <XBee.h>

// defining the pin where we connect the led

const int ledPin = 13;

//setup routine

void setup()

{

pinMode(ledPin, OUTPUT);// making the ledpin an outout

}

// repeat forever

void loop()

{

digitalWrite(ledPin,HIGH);// led on

delay(500);// waiht 500ms

digitalWrite(ledPin,LOW);// led of

delay(500);

}

Opdracht 2 : Analog read

Code:

void setup() {

// initialize serial communication at 9600 bits per second:

Serial.begin(9600);

}

// the loop routine runs over and over again forever:

void loop() {

// read the input on analog pin 0:

int sensorValue = analogRead(A0);

// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):

float voltage = sensorValue \* (5.0 / 1023.0);

// print out the value you read:

Serial.println(voltage);

}

Verbeterde code:

void setup() {

// initialize serial communication at 9600 bits per second:

Serial.begin(9600);

}

// the loop routine runs over and over again forever:

void loop()

{

float SensorValue = 0;// declararation and init of the variable

SensorValue= SensorRead();// Reading the return value of the function

Serial.println(SensorValue);// printing the sensor value in volts

}

float SensorRead()

{

// read the input on analog pin 0:

int sensorValue = analogRead(A0);

// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):

float voltage = sensorValue \* (5.0 / 1023.0);

// return the value we need

return voltage;

}

Opdracht 3 : xbee broadcasting

#include <Printers.h>

#include <XBee.h>

XBee xbee = XBee();// creating a xbee class

uint8\_t payload[] = "Ben De Lathouwer";// what do we whant to send

XBeeAddress64 addr64 = XBeeAddress64(0x0, 0x0000ffff);

ZBTxRequest zbTx = ZBTxRequest(addr64, payload, sizeof(payload));// make Tx packet

void setup()

{

Serial.begin(9600);// init of the serial port

xbee.setSerial(Serial); // Tell XBee to use Hardware Serial.

}

void loop()

{

xbee.send(zbTx); // Send your request

delay(2000);// delay for 2 seconds

}

kXBeeAddress64 addr64 = XBeeAddress64(0x0, 0x0000ffff);

In dit stukje code schrijven we het broadcast adders van de ontvanger.

ZBTxRequest zbTx = ZBTxRequest(addr64, payload, sizeof(payload));// make Tx packet

In de bovenstaande stukje code maken we een packet dat we gaan verzenden. En deze is afhankelijk van de groote van de payload

Labo 4A

// Opdracht 4A: Unicast

// Maak zend programma. Zodat zender om de 2 sec uw naam stuuurt

// naar de onvanger.

// Adress Xbee: 0013A200 406F4B31(ontvanger)

// bibliotheek toevoegen aan code

#include <XBee.h>

// aanmaken van een object Xbee

XBee xbee = XBee();

// maken van een string met de doortesturen informatie

uint8\_t payload[] = "Ben De Lathouwer";

// instellen van het zendt adress SH + SL

XBeeAddress64 addr64 = XBeeAddress64(0x0013A200, 0x408BEE26);// opgegeven adderes maakt dit unicast

// maken van een TX verzoek

ZBTxRequest zbTx = ZBTxRequest(addr64, payload, sizeof(payload));

void setup() {

Serial.begin(9600); // instellen seriele communicatie

xbee.begin(Serial); //

}

void loop() {

// zenden van het verzoek

xbee.send(zbTx);

// delay van 2 seconden

delay(2000);

}

XBeeAddress64 addr64 = XBeeAddress64(0x0013A200, 0x408BEE26);// opgegeven adderes maakt dit unicast

In dit stukje coden geven we haet adderes mee van de xbee die onze boodschap moet ontvangen

4B

// Opdracht 4A: Unicast

// Maak ontvangst programma.

// Adress Xbee: 0013A200 406F4B31

// bibliotheek toevoegen aan code

#include <XBee.h>

// aanmaken van een object Xbee

XBee xbee = XBee() ;

// maken van de klasse response

XBeeResponse response = XBeeResponse() ;

// maken van de klasse rx

ZBRxResponse rx = ZBRxResponse() ;

// stelt een 64 bit getal voor

XBeeAddress64 raddr64 ;

// 32 bit integer adres\_lsb

uint32\_t adres\_lsb ;

// 32 bit integer adres\_msb

uint32\_t adres\_msb ;

void setup() {

Serial.begin(9600) ; // instellen van baut rare op 9600

xbee.setSerial(Serial) ;

}

void loop() {

// opvangen van de data

xbee.readPacket() ;

// leest alle ontvangen data

if (xbee.getResponse().isAvailable()) {

// de if functie is vervuld als er een pakeket word ontvangen

if (xbee.getResponse().getApiId() == ZB\_RX\_RESPONSE) {

// de klasse rx word gevuld met de data

xbee.getResponse().getZBRxResponse(rx) ;

// kijken naar het adres van de ontvanger

raddr64 = rx.getRemoteAddress64() ;

// aders opsplitesen in de 2 variabelen

adres\_msb = raddr64.getMsb() ;

adres\_lsb = raddr64.getLsb() ;

// als het adres van de ontvanger gelijk aan het adres het

//ingestelde adres is het statmen voldaan

//if ((adres\_msb == 0x0013A200 ) && (adres\_lsb == 0x406F4B3D))

//{

// bepaald de lengte van het paket en print het af op de seriële monitor

for (int i = 0; i < rx.getDataLength(); i++) {

Serial.print((char) rx.getData(i)) ;

// }

}

Serial.println() ;

}

adres\_msb = raddr64.getMsb() ;

adres\_lsb = raddr64.getLsb();

in deze stukjes coden splitsen we het aderres van de zender op in een Ms ben een lsb

in dit stukje code wordt de lengte van het paket bepaald en afgeprint op de seriële monitor

for (int i = 0; i < rx.getDataLength(); i++) {

Serial.print((char) rx.getData(i)) ;

kijken naar het adres van de ontvanger

raddr64 = rx.getRemoteAddress64() ;

Labo 5

#include <Printers.h>

#include <XBee.h>

// Opdracht5 Unicast zend en ontvangen

// Maak zend programma. Zodat zender om de 2 sec uw naam stuuurt

// naar de onvanger.

// Adress Xbee: 0013A200 406F4B31(ontvanger)Radek

//0x0013A200, 0x406F4B3D nico ontvanger

#define TIMEOUT 4000

// aanmaken van een object Xbee

XBee xbee = XBee();

// maken van een string met de doortesturen informatie

uint8\_t payload[] = "testBen";

// instellen van het zendt adress SH + SL

XBeeAddress64 addr64 = XBeeAddress64(0x0013A200, 408BEE26);// opgegeven adderes maakt dit unicast

// maken van een TX verzoek

ZBTxRequest zbTx = ZBTxRequest(addr64, payload, sizeof(payload));

// maken van de klasse response

XBeeResponse response = XBeeResponse() ;

// maken van de klasse rx

ZBRxResponse rx = ZBRxResponse() ;

// stelt een 64 bit getal voor

XBeeAddress64 raddr64 ;

// 32 bit integer adres\_lsb

uint32\_t adres\_lsb ;

// 32 bit integer adres\_msb

uint32\_t adres\_msb ;

void zend(void);

void ontvang(void);

void setup()

{

Serial.begin(9600); // instellen seriele communicatie

xbee.begin(Serial); //

}

void loop()

{

ontvang();

zend();

}

void zend (void)

{

// zenden van het verzoek

xbee.send(zbTx);

// delay van 2 seconden

delay(2000);

}

void ontvang(void)

{

if (xbee.readPacket(TIMEOUT)) {

// leest alle ontvangen data

if (xbee.getResponse().isAvailable())

{

// de if functie is vervuld als er een pakeket word ontvangen

if (xbee.getResponse().getApiId() == ZB\_RX\_RESPONSE)

{

// de klasse rx word gevuld met de data

xbee.getResponse().getZBRxResponse(rx) ;

// kijken naar het adres van de ontvanger

raddr64 = rx.getRemoteAddress64() ;

// aders opsplitesen in de 2 variabelen

adres\_msb = raddr64.getMsb() ;

adres\_lsb = raddr64.getLsb() ;

// bepaald de lengte van het paket en print het af op de seriële monitor

for (int i = 0; i < rx.getDataLength(); i++)

{

Serial.print((char) rx.getData(i)) ;

}

Serial.println() ;

}

}

}

Code is het zelfde als labo 4a en 4b alleen heb ik deze lijn code moeten toevoegen

if (xbee.readPacket(TIMEOUT))

{

…

}

Deze stopt de functie als de timout tijd bereikt is

Labo 6

#include <Printers.h>

#include <XBee.h>

// aanmaken van een object Xbee

XBee xbee = XBee() ;

// maken van de klasse response

XBeeResponse response = XBeeResponse() ;

// maken van de klasse rx

ZBRxResponse rx = ZBRxResponse() ;

// stelt een 64 bit getal voor

XBeeAddress64 raddr64 = XBeeAddress64();

// 32 bit integer adres\_lsb

uint32\_t adres\_lsb ;

// 32 bit integer adres\_msb

uint32\_t adres\_msb ;

void setup()

{

Serial.begin(9600);// init of the serial port

xbee.setSerial(Serial); // Tell XBee to use Hardware Serial.

}

void loop()

{

Serial.println(rx.getDataLength ()); // datalengte van de transmitter ontvangen

Serial.println(rx.getOption ()); // option veld van de transmitter ontvangen(hiermee kun je zien of de zender unicast of broadcast zend)

Serial.println (int(rx.getRemoteAddress64())); // adres van de transmitter ontvangen

Serial.println (rx.getRemoteAddress16 ()); // 16 bit adres van de transmitter ontvangen

Serial.println (raddr64.getMsb ());

Serial.println (raddr64.getLsb ());

}

Serial.println (int(rx.getRemoteAddress64())); om het remote 16 bit adderes af te drukken moeten we typcasten naar een integer.Dit zodat onze serial print er mee overweg kan