Arbeitsblatt digitale Ein- und Ausgabe mit: Bluetooth, UART => HC05

HC05 Bluetooth-Modul



Anschluss: (Baseshield)

State PA\_1

RxD PB\_11

TxD PB\_10

GND

5V

Enable PB\_12

App: Z. B. Bluetooth Electronics

**UART Instruktionen**

**Deklaration:**

BufferedSerial *name*(PinName tx, PinName rx, int [baud](https://os.mbed.com/docs/mbed-os/v6.2/mbed-os-api-doxy/classmbed_1_1_serial_base.html#a9afb7aa9321cd71a8a26a673157583d2))

z.B. Deklaration einer UART-Schnittstelle mit den Anschlüssen tx=PB\_10, rx=PB\_11, Übertragungsgeschwindigkeit 9600 Bit/s:

BufferedSerial hc05(PB\_10,PB\_11,9600);  


**Daten empfangen:**

Int *anz=Name*.read(char\* daten,int length)

Mit char\* daten: Char-Array zur Aufnahme der Daten

Int length: Maximale Bytezahl

Return int anz: Anzahl der empfangenen Datenbytes

z.B.:

char daten[16];

int anz;

anz = hc05.read(daten,16);

**Daten senden:**

Int *anz=Name*.write(char\* daten,int length)

Mit char\* daten: Char-Array mit den Versendedaten

Int length: Maximale Bytezahl zum versenden

Return int anz: Anzahl der versendeten Datenbytes

z.B.:

char daten[16];

int anz;

anz = hc05.write(daten,16);

**Prüfen ob Daten verfügbar sind:**bool verfuegbar=hc5.readable();

**Prüfen ob Daten versendbar sind:**

bool versendbar=hc5.writeable();

**Oberfläche bei Bluetooth-Electronics:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Handy sendet,  HC05 empfängt | Aktion | Bluetooth Oberfläche | HC05 sendet,  Handy empfängt | Bei Taste |
| Turn On: ‚A‘  Turn Off: ‚a‘ | PC\_0 =1  PC\_0 =0 |  | \*LR255G0B0\*  \*LR0G0B0\* | PB\_0=1  PB\_0=0 |
| Turn On: ‚B‘  Turn Off: ‚b‘ | PC\_1 =1  PC\_1=0 | \*MR0G255B0\*  \*MR0G0B0\* | PB\_3=1  PB\_3=0 |
| Turn On: ‚C‘  Turn Off: ‚c‘ | PC\_2=1  PC\_2=0 | \*NR0G0B255\*  \*NR0G0B0\* | PB\_4=1  PB\_4=0 |
| Turn On: ‚D‘  Turn Off: ‚d‘ | PC\_3=1  PC\_3=0 |  |  |

**Aufgaben**

1. Erstellen Sie die Bedienoberfläche mit Bluetooth Electronics (o. ä.) und vervollständigen Sie das Programm:

//Blinkingrate in milliseconds

#define BLINKING\_RATE    20ms

BufferedSerial hc05(PB\_10,PB\_11,9600);

DigitalOut a(PC\_0);

…

DigitalIn l(PB\_0);

…

int main()

{

char daten[15];

 l.mode(PullDown);

 int altL=0, altM=0, altN=0;

    while (true) {

        if (l==1&&altL==0)hc05.write("\*LR255G0B0\*",11);

…

       altL=l;

…

        if (hc05.readable())

        {

            ThisThread::sleep\_for(BLINKING\_RATE); //Alle Daten abwarten

            hc05.read(daten,1);

            if (daten[0]=='A') a=1;

…

            if (daten[0]=='a') a=0;

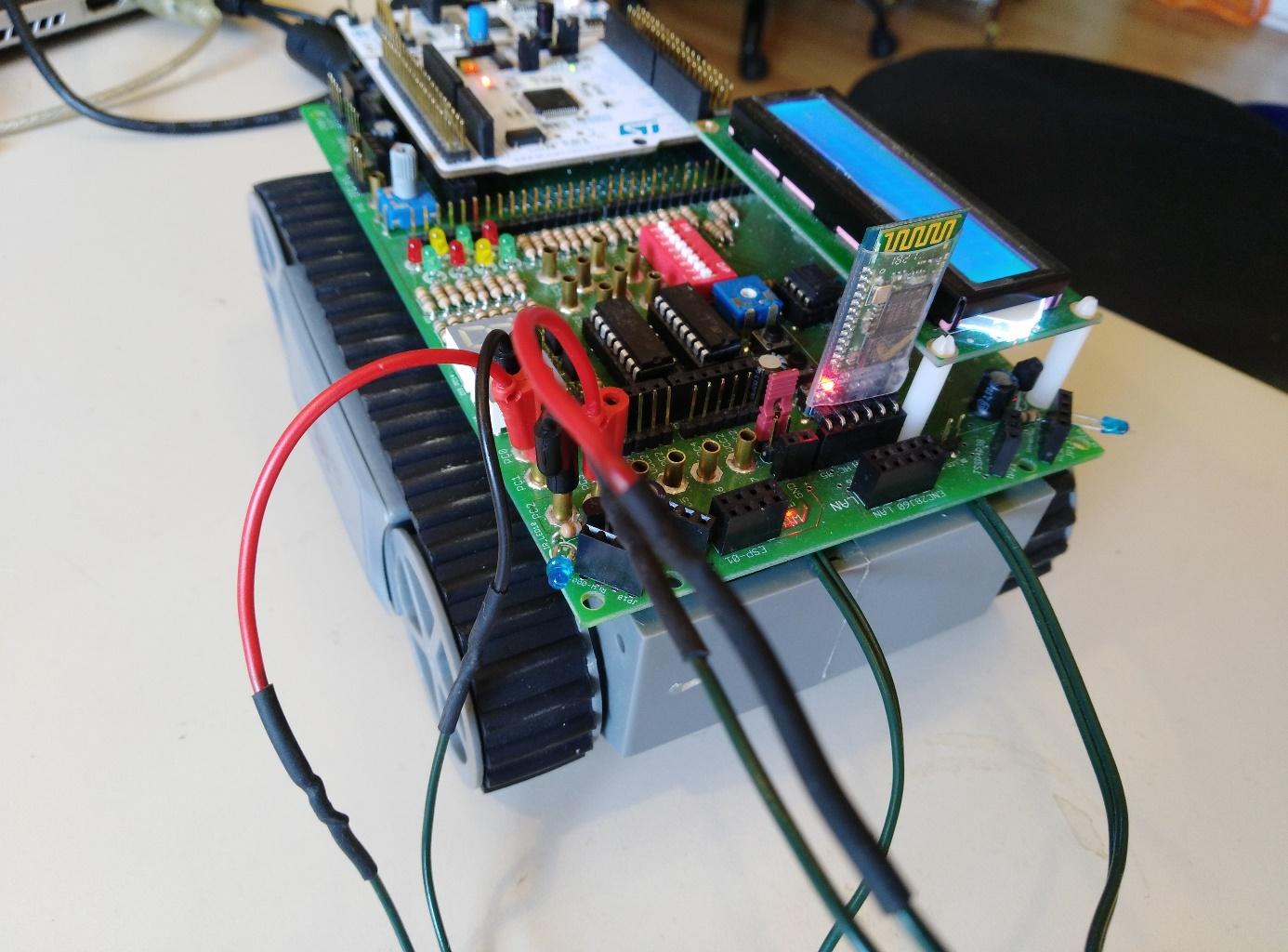
…

        }

        ThisThread::sleep\_for(BLINKING\_RATE);  //entprellen

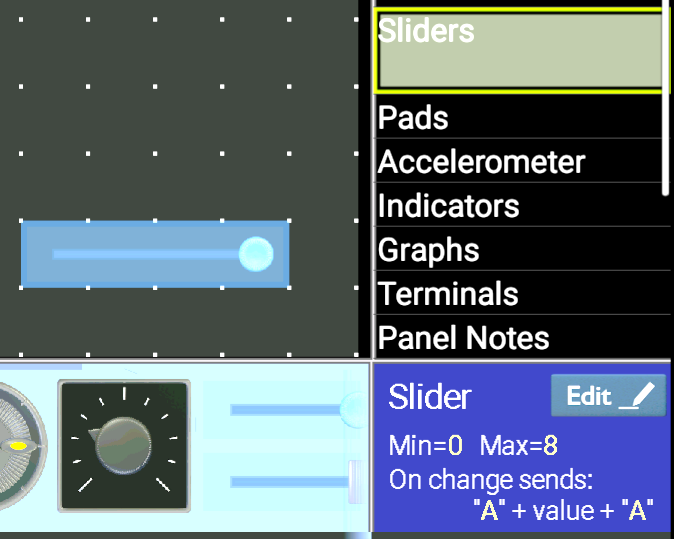
    }

}

1. Roboterfahrzeug steuern mit Bluetooth

Schließen Sie das Roboterfahrzeug an PC\_0 .. PC\_3 Hochstromausgänge an und optimieren Sie die Steuerung

1. Balkenanzeige

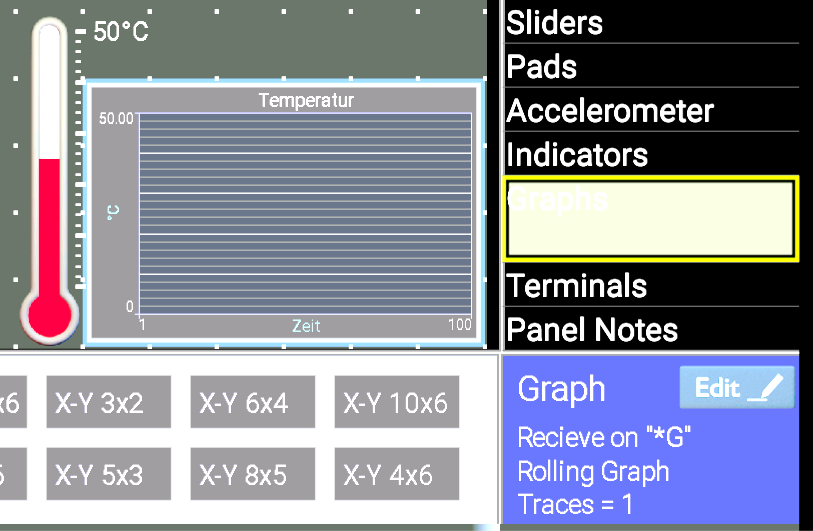
Benötigt wird ein Slider, der die Werte 0 .. 8 versenden kann. Das Handy sendet, wenn der Slider bewegt wird die Botschaft ‚A‘+Wert+‘A‘

Der Slider soll eine Balkenanzeige an PC\_0 .. PC\_7 steuern.

Lösungshinweise:

* Wenn Zeichen vorhanden sind (readable)
  + Zunächst 5ms warten, um dem UART genügend Zeit zu geben um die komplette Botschaft zu übertragen
  + dann 1 Zeichen holen mit: hc05.read(daten,1);
  + Wenn daten[0]=‘A‘ dann
    - Weitere 2 Zeichen holen mit hc05.read(daten,2);
    - Den Zahlenwert aus dem char-Array extrahieren mit: sscanf(daten,"%d",&wert);
    - Den Zahlenwert in Form eines Balkens ausgeben

1. Temperaturanzeige

Benötigt wird ein Graph.

Zur Anzeige der Temperatur muss der Mikrocontroller folgende Daten schicken:

„\*G“+Wert+“\*“

Das Thermometer ist unter „Indicators“ zu finden und ist für die Botschaft „\*T“+Wert+“\*“ zu konfigurieren.

Vervollständigen Sie den Code

#define BLINKING\_RATE 500ms

BufferedSerial hc05(PB\_10,PB\_11,9600);

AnalogIn ain2(PA\_4);

lcd mylcd;

int main()

{

char daten[15];

float Wert = 0; // variable to store the value read

float R2\_25 =1500;

float R2\_theta;

float R1=1500;

float dt;

float t;

float alpha=-0.045;

while (true) {

/\*Temperaturberechnung\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

R2\_theta=R1\*ain2/(1-ain2);

dt=(R2\_theta/R2\_25-1)/alpha; //näherungsweise

t=25+dt;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Ihr Code

ThisThread::sleep\_for(BLINKING\_RATE);

}

}

Hinweis:

Mit: sprintf(daten,"\*G%d\*",(int)t); kann der Sendestring zusammengebaut werden.