//Arduino\_HTC05\_HardwareSerial\_Monitor.ino

//Board: Arduino

//Bluetoothmodul:BT Modul HTC05

//Serielle Schnittstelle: HardwareSerial UART

//Monitor

// Steuerung über ein Eingabe auf dem Monitoranwendung der LGI-App

// Hier mit den Befehlen Serial.print und Serial.read

Zuerst überprüfen wir unsere Hardware: Welches Board willst du einsetzen: Arduino oder Attiny? und welches Bluetoothmodul verwendest du ein HTC05-Modul oder das Grove Modul?

Für den Attiny gibt es keinen eigenen Block. Man muss nur beachten, dass er kein Hardware-Serial hat und dass man beim Softwareserial die Pins 0 und 1 nimmt.

Mit dem Arduino kann über die eingebaute serielle Schnittstelle UART kommunizieren oder über die softwareseitig aufgebaute Schnittstelle "software-Serial". Mit dem Attiny hat keine Hardware-Schnittstelle (kein UART). Man muss dort die software-Serial-Schnittsstelle verwenden.

Die beiden Anwendungen der LGI-App "Monitor" und "Fernbedienung" (Remote) stehen jeweils für beide Boards, beide Bluetoothmodule und beide seriellen Kommunikationswege zur Verfügung.

Unterschied zwischen HW

Die erste Festlegung muss noch vor dem Setup getroffen werden: Welche serielle Schnittstelle wird eingesetzt?

a) Soll die UART: Hardwareschnittstelle genutzt werden, dann Aufruf einer Instanz mit dem Befehl: Android Remote(); oder Android Monitor();ohne Angabe der Rx und Tx – Pins. Die Kommunikation läuft harwaremäßig auf Pin0 und Pin1. Der Attiny hat keine UART-schnittstelle!

b) Beim Arduino ist man frei. Man kann über die UART-Schnittstelle (siehe Punkt a)) kommunizieren oder über die Softwareschnittstelle software.Serial. Der Attiny kann nur über die Software-Schnittstelle kommunizieren. Der Aufruf der Instanz mit dem Befehl: Android Remote(4,5); bzw. Android Monitor(4,5); Du kannst (fast) frei wählen, über welche Pins die Kommunikation laufen soll. Nicht nehmen solltest du beim Arduino die Pins 0 und 1, da diese beiden Pins für die UART-schnittstelle reserviert sind. Für den attiny bietet sich dagegen Pin0 und Pin1 an.

c) Soll die Kommunikation beim Arduino über die UART-Schnittstelle laufen, dann kann man die serial.Event() nutzen. Normalerweise muss man im Programm ständig abfragen, ob neue Werte an der Schnittstelle anliegen. Das nennt man „pollen“. Du läufst also ständig zum Briefkasten und schaust nach, ob Post da ist. Wenn Post da ist, dann musst du den Briefkasten leeren, sonst quillt er über. Bei serial.event kannst du dir das Nachsehen sparen. Der Postbote klingelt, wenn er etwas einwirft. Du musst also den Briefkasten nur leeren, wenn es geklingelt hat.

Beim Einsatz von Bluetooth brauchst du diese Funktion nur bei der Fernsteuerung (Remote). Bei der Steuerung über den Monitor musst du pollen.

Sobald sich an der Schnittstelle etwas tut wird die Funktion serialEvent() aufgerufen. Die Funktion liest dann die Information ein: ein ASCii-Zeichen also ein Zeichen auf deiner Tastatur. Deine Kommunikation muss bei SerialEvent über Pin0 und Pin1 gehen!.

void serialEvent() {Remote.readRemote();}

// Die Festlegung muss im Setup getroffen werden: Mit wie viel Baud wird kommuniziert?

// a) Remote.BTserial\_begin(9600); bei der UART-Schnittstelle

// b) Remote.BTsoftSerial\_begin(9600); bei einer softseriellen Schnittstelle

// c) Remote.BTGrove\_begin(9600); bei einer seriellen Schnittstelle mit Grovemodul

// d) Remote.BTsoftGrove\_begin(9600); bei einer softseriellen Schnittstelle mit Grovemoduel

// In der loop lauten dann die Befehle (egal ob Grove oder HTC05)

// a) Remote.readMonitor oder Remote.readSoftMonitor

// b) Remote.printMonitor oder Remote.printSoftMonitor

// c)

// d)

// für den Seriel event muss folgendes

// Falls Arduino: Bei tools das passende Arduino-Modul wählen

// Falls HTC05 Bluetooth - Modul

// eingebaute UART-Schnittstelle

// falls software-Serial-Schnittstelle

// Falls Fernsteueurng - Remote

// Falls Monitor-Steuerung

// Falls Attiny:

// Falls Grove Bluetooth- Moduel