

Aufgabenstellung für die Ausarbeitung Netzwerke SS2019

Hinweise zur Abgabe:

- Dokumentieren Sie Ihren Code
- Erstellen Sie eine zip oder rar Datei mit dem Visual Studio Projekt ohne der Datei *.sdf.
- Senden Sie auch ein Wireshark-Capturefile. Das Wireshark File soll den Durchlauf Ihres Programms zeigen
- Die Datei senden Sie an die Mailadresse parzhuber@hm.edu
- Die Aufgaben können in Gruppenarbeit mit maximal 3 Teilnehmern gelöst werden. Abgabe der Arbeiten ist eine Woche vor Beginn der Prüfungen.

Aufgabenstellung 1: Daytime – Server und Client

Erstellen Sie zwei Visual Studio Projekte: ein Projekt für den Daytime-Server (*server_daytime*) und eines für den Daytime-Client (*client_daytime*).

- Der Server soll nur auf Anfragen an den für das daytime Protokoll reservierten Port antworten. Selbstverständlich soll er jedoch von beliebigen Clients die Anfragen beantworten.

```
//Codeschnippel für den Zugriff auf die Rechnerzeit:  
  
time_t rawtime;  
string timey = ctime(&rawtime); //liefert formatiert: Www Mmm dd hh:mm:ss yyyy  
// Den string zerlegen  
string day = timey.substr(0,3);  
. . .
```

- Der Client soll sich von dem Server die Zeit holen und auf der Konsole im folgenden Format darstellen:

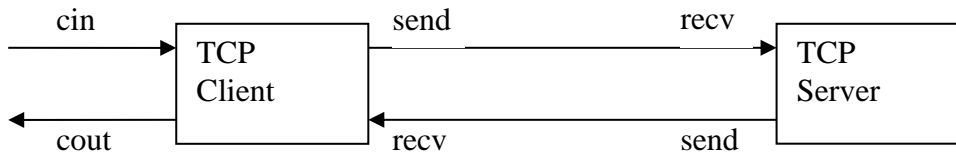
Der Server mit der Adresse `??.??` liefert die folgende Zeit:

```
Datum:      23. April.2016  
Uhrzeit:    9:43
```

- Alternativ soll es der Server auch ermöglichen, die Zeit von einem gewöhnlichen Browser aus abzufragen.
- Die Zeit soll in derselben Formatierung wie oben mittig in großer Schrift im Browser angezeigt werden.

Aufgabenstellung 2: Client-Server Beispiel mit TCP

Das folgende Bild zeigt ein Beispiel für einen Echo-Client und einen Echo-Server:



Erstellen Sie zwei Visual Studio Projekte, eines für den Client und eines für den Server.

- Client *client_example* und
- Server *server_example*

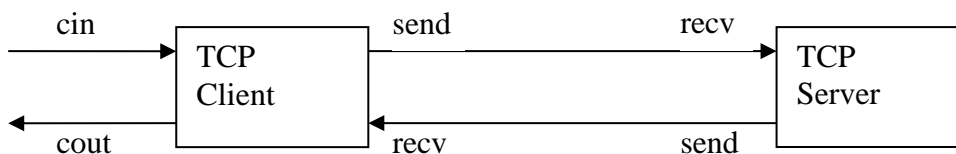
Der Client liest beliebige Eingaben von der Konsole und schickt die Daten an einen Server. Der Server liest die Daten und schickt sie wieder zu dem Client zurück. Der Client gibt diese eingegebenen ASCII Zeichen wieder an die Konsole aus. Selbstverständlich sollen diese Daten auch auf der Serverkonsole angezeigt werden.

Wenn Sie nun an der Client Konsole Zeichen eingeben sollten diese unverstümmelt wieder auf derselben Konsole ausgegeben werden (wie ein Echo...)

Optional sollen die eingegebenen Zeichen erst nach der Eingabe eines Zeilenumbruchs als Echo zurückgeschickt werden.

Aufgabenstellung 3: Client-Server Beispiel mit UDP

Das folgende Bild zeigt ein Beispiel für einen Echo-Client und einen Echo-Server:



Erstellen Sie zwei Visual Studio Projekte, eines für den Client und eines für den Server.

- Client *client_example* und
- Server *server_example*

Der Client liest beliebige Eingaben von der Konsole und schickt die Daten an einen Server. Der Server liest die Daten und schickt sie wieder zu dem Client zurück. Der Client gibt diese eingegebenen ASCII Zeichen wieder an die Konsole aus. Selbstverständlich sollen diese Daten auch auf der Serverkonsole angezeigt werden.

Wenn Sie nun an der Client Konsole Zeichen eingeben sollten diese unverstümmelt wieder auf derselben Konsole ausgegeben werden (wie ein Echo...)

Optional sollen die eingegebenen Zeichen erst nach der Eingabe eines Zeilenumbruchs als Echo zurückgeschickt werden

Aufgabenstellung 5: Client-Server Beispiel mit TCP Socket

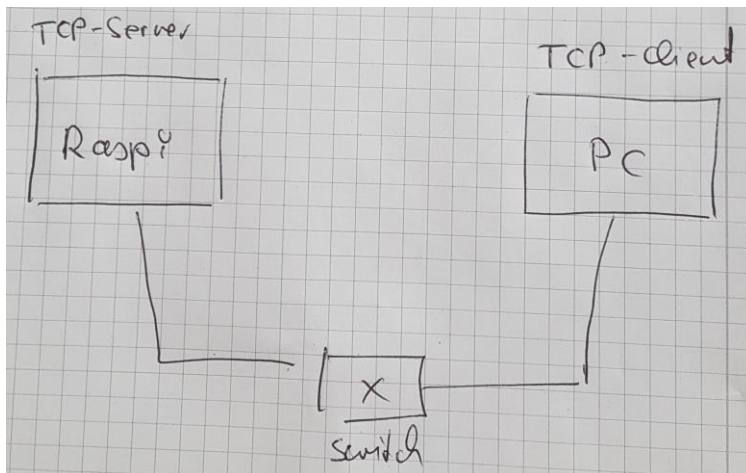
Programmieren Sie einen TCP-Client auf dem PC und einen TCP Server auf dem RaspberryPi. Auf der Konsole des TCP-Servers soll ausgegeben werden:

„Bind und Listen erfolgreich“
„Warte auf Verbindung mit PC Client“

Sobald eine Verbindung erfolgt ist soll ausgegeben werden:

„Verbindung mit Rechner an Port xxx erfolgreich“
„Warte auf Daten:“

Sobald Daten vom Client gesendet werden sollen diese angezeigt werden.



Aufgabenstellung 4: Client-Server Beispiel mit UDP Socket und RasperryPi

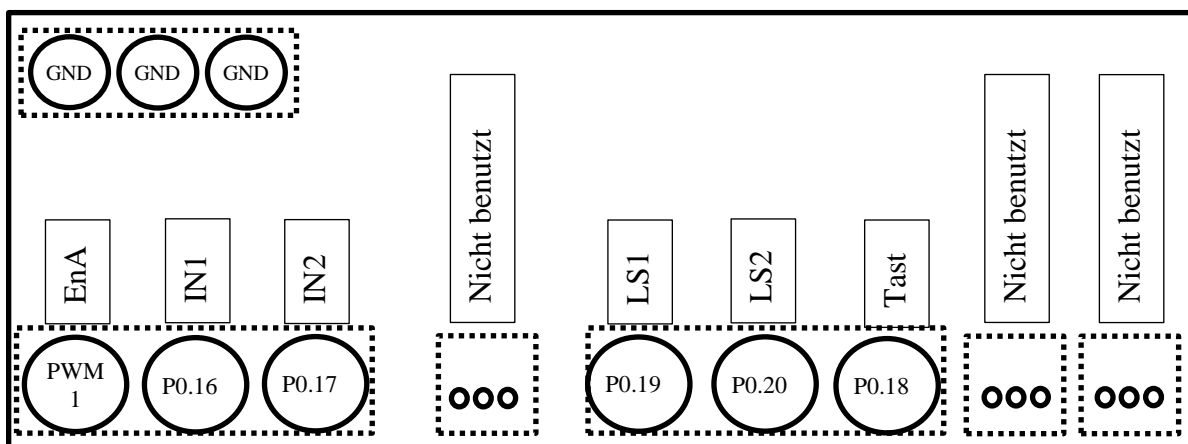
Gegeben ist das folgende Förderband:



Die Anschlüsse zeigt das folgende Bild:

	Port.Pin	Beschreibung
Motoransteuerung:	GPIO.x	„1“ = Band läuft, „0“=Band steht
	GPIO.x	1/0 = links bzw. 0/1 = rechts
	GPIO.x	
Taster/Impulszähler	GPIO.x	Offener Taster = +5V, active low
Lichtschanke	GPIO.x	Lichtschanke 1, active low
	GPIO.x	Lichtschanke 2, active low

Verkabelung der Adapterplatine mit dem Praktikumsboard:



- Belegen Sie die GPIOs nach Ihren Wünschen.

Client:

Am einfachsten schreiben Sie einen Client auf dem PC, Sie können aber auch einen Browser verwenden.

Server:

Als Server fungiert ein Raspi. Die Befehle um das Förderband anzusteuern sind wie folgt:

- „StartL“: Starten des Förderbandes nach links
- „StartR“: Starten des Förderbandes nach rechts
- „Stop“: Stoppen des Förderbandes

Bei Erreichen der Lichtschranke soll das Förderband die Richtung umschalten