Remotrol von FabianE.

Im Roboternetz hatte FabianE. eine recht umfangreiche Fernsteuerung präsentiert, welche in Version 1.3 recht gut lauffähig war und sehr viel abdeckte und auch mit allen Erweiterungen kombiniert werden konnte. Den damaligen Forumseintrag gibts HIER.

fabqu hat die Version 1.3 bei sich und wird diese, falls FabianE. es erlaubt, erneut in etwas ageänderter Variante veröffentlichen.

Die Firmware auf dem RP6 war als "Slave" gedacht und erwartete im Wesentlichen über Bluetooth Befehle vom PC und schickte Sensordaten darüber an den PC. Dabei kommuniziert der RP6 über seine UART-Schnittstelle (universal asynchronous receive/transmit), also Rx (receive data) und Tx (transmit data). Je nachdem, welche Plattform (RP6-Base, M32, M128 oder M256) man verwendet, werden entsprechend alle verfügbaren Sensordaten routinemäßig über die UART-Schnittstelle gesendet. Die `Baudrate ist dabei 38.400`. Die Befehlsstruktur, welche der RP6 erwartet, ist etwas komplizierter.

Senden von Befehlen an den RP6

Die Kommandos sind vom Typ `#E1:(V1):(V2):E2:id*` -> Beginn ist `#`, Ende ist `*`. `id` ist ein Counter, der stets inkrementiert werden muss und maximal 99 sein darf. `E1` und `id` sind erforderlich. `E2` ist obligatorisch. `V1` und `V2` sind weitere Parameter, die es nicht immer braucht.

Erste Ebene	E1	Zweite Ebene	E2
#define CMD_SET_SPEED	1	Features:	
#define CMD_SET_SERVO	2	#define SET_FEATURE_GENERAL	0
#define CMD_SET_LEDS	3	#define SET_FEATURE_LIGHT	1
#define CMD_SET_BEEP	4	#define SET_FEATURE_INT0	3
#define CMD_SET_START_MELODY	5	#define SET_FEATURE_MIC	4
#define CMD_SET_FEATURE	6	#define SET_FEATURE_SRF08_RADAR	5

Erste Ebene	E1	Zweite Ebene	E2
#define CMD_SET_STOP	7	#define SET_FEATURE_SRF02	6
#define CMD_SET_CONNECTION_SPEED	8		
#define CMD_GET_FIRMWARE	9	LEDs:	
#define CMD_SET_ACSPOWER	10	#define LEDS_RP6	0
#define CMD_SET_TEST	11	#define LEDS_M32	1
#define CMD_RESET_ID_COUNTER	99		
x		ACS:	
x		#define ACS_POWER_OFF	0
x		#define ACS_POWER_LOW	1
x		#define ACS_POWER_MED	2
x		#define ACS_POWER_HIGH	3
x			
x		Tests	
x		#define TEST_LCD	0
x		#define TEST_BEEPER	1
x		#define TEST_LED	3
x		#define TEST_EXTERNAL_MEMORY	4
x		#define TEST_I2CLED	5
x		#define TEST_I2CMOTOR	6
x		#define TEST_MIC	7
x		#define TEST_MOTOR	8
x		#define TEST_BATTERY	9
x		#define TEST_ACS	10
x		#define TEST_BUMPER	11
x		#define TEST_LIGHTSENSOR	12

Beispiele:

- `#7: 21*` -> Stop, mit id=21
- `#1: 50: 75: 0: 22*` -> fahren (E1=1) vorwärts (E2=0) mit Geschwindigkeit V1=50 linke Kette, V2=75 rechte Kette (von 200) und id=22
- `#3: 0: 33: 23*` -> LED (E1=3) auf der RP6-Base (E2=0), es warden die Status LEDs SL1 (Bit 1=1=1) und SL6 (Bit 6=1=32) angeschaltet (V1=1+32=33)

Empfangen von Daten vom RP6

Die Daten des RP6 kommen in Blöcken zu mehr oder weniger festen Zeitintervallen. Die Blöcke sehen wie folgt aus:

```
[DATA]
Bat: 955
SpeedL: 0
SpeedR: 0
PowerL: 0
PowerR: 2
LightL: 592
LightR: 865
BumpL: 0
BumpR: 0
ObsL: 1
ObsR: 1
ADC0: 860
ADC1: 855
BumpHL: 0
BumpHR: 0
[/DATA]
```

Sie beginnen also mit `[DATA]` und enden mit `[/DATA]`, dazwischen stehen die Werte in je einer Zeile in Tupeln aus einem Keyword und dem Wert; beide sind durch Doppelpunkt `:` getrennt. Beim Senden wird kein Interrupt o.ä. ausgelöst, man muss also auf der anderen Seite stets horchen und den Puffer bei Bedarf lesen.

Revision #5
Created 19 November 2022 17:10:36 by Fabian
Updated 19 March 2024 13:42:26 by Guest