

Molch Kommunikation

Stand: 20.Mai.2008

1 Kommunikation Modes

Es wird zwischen 3 verschiedenen Kommunikationsmodes unterschieden.

1. Mode Idle
In diesem Mode wartet der ARM auf ein 2 Byte Kommando vom PIC Prozessor.
2. Mode Pipe
In diesem Mode führt der ARM Signalberechnungen durch, schreib jede Sekunde 1KB an Ergebnissen auf die SD-Karte. In diesen 1KB sind auch 17 Byte Sensor-Daten des PICs enthalten, die dieser sekundlich nach dem Empfang des Alive Bytes [0x5A] zum ARM schickt.
3. Mode USB
In diesem Mode hat der ARM vom PC über die USB Schnittstelle 32KB Daten erhalten und zum PIC durchgeleitet, der wartet dann auf die 130 Byte Antwort des PIC und sendet diese weiter zum PC. Danach kehrt der ARM automatisch in den Modus IDLE zurück.

2 Serielle Kommunikation zwischen ARM & PIC

2.1 2 Byte Kommandos im Idle Mode (PIC → ARM)

- START [0x827D]
Der ARM antwortet mit einem ECHO des Befehls und wechselt dann in den Mode Pipe.
- CONTIN [0x837C]
Der ARM antwortet mit einem ECHO des Befehls und wechselt dann in den Mode Pipe. Die Aufzeichnung wird an der Position auf der SD-Karte fortgesetzt, die in der Settings.dat Datei bei jedem Stop Kommando abgelegt wird.

2.2 17 Byte Kommandos im Pipe Mode (PIC → ARM)

Diese Kommandos werden sekundlich vom PIC gesendet, wenn sich der ARM im Pipe-Modus befindet. Jede Sekunde erhält der ARM 17 Byte vom PIC, wovon die ersten zwei Bytes den Befehl festlegen und die übrigen 15 Bytes Platz für (Sensor)Daten bieten.

- STOP[0x827D]
Der ARM stoppt die Aufzeichnung, speichert die aktuelle Schreibposition auf der SD-Karte und wechselt dann in den Idle-Modus. Die übrigen 15 Bytes sind ohne ohne Bedeutung, müssen aber mitgeschickt werden.
- DATA [0xC03F]
Dieser Befehl übergibt dem ARM die 15 Byte an Sensor-Daten des PICs, die für die laufende Sekunde mit abgespeichert werden soll. Die übrigen 15 Bytes beinhalten also die Sensor-Daten des PICs.

2.3 1 Byte Kommandos im Pipe Mode (ARM → PIC)

Jede Sekunde sendet der ARM 1Byte [0x5A] an den PIC zu zum Zweck der Synchronisation. Der PIC sendet nach dem Empfang des Alive-Bytes das nächste 17-Byte Kommando.

3 USB Kommunikation zwischen PC & PIC

Wenn der ARM sich im IDLE-Modus befindet, kann der Molch jeder Zeit über die USB-Schnittstelle mit einem PC verbunden werden. Die Daten der SD-Karte werden als Massenspeicher zur Verfügung gestellt. Soll darüberhinaus mit vom PC mit dem PIC kommuniziert werden, wird folgendes Protokoll mit festen Nachrichtenlängen verwendet.

- Nachrichtenlänge PC → PIC (Anfrage): 2Byte CMD + 30Byte Daten, 32 Bytes
- Nachrichtenlänge PIC → PIC (Antwort): 2Byte CMD + 128Byte Daten, 130 Bytes

Wenn der PC mit dem PIC kommunizieren möchte schickt er ein 32 Byte Kommando über USB zum ARM. Der Arm leitet das Paket uninterpretiert an den PIC weiter und wechselt in den USB-Mode. Der PIC antwortet indem er eine 130 Byte Antwort an den ARM zurückschickt. Das 2Byte Kommandofeld in der Antwort ist ein Echo des gleichen Feldes in der Anfrage. In den übrigen 128 Byte wird die Antwort des PICs verpackt, nicht benötigte Bytes werden auf [0xDD] (Dummy) gesetzt. Nach dem Empfang der Antwort vom PIC und der Weiterleitung zum PC wechselt der ARM zurück in den Idle-Modus.

3.1 Anfragen des PCs

1. Get_codemld:
 - 2Byte Anfrage CMD: [0x00FF]
 - 30 Byte Anfrage: Dummys [0xDD]
 - 2 Byte Antwort CMD: [0x00FF]
 - 128 Byte Antwort: Gerätenr, Pgr.-Version, etc, + Dummys [0xDD]
2. Get_rampic:
 - 2Byte Anfrage CMD: [0x05FA]
 - 30 Byte Anfrage: Dummys [0xDD]
 - 2 Byte Antwort CMD: [0x05FA]
 - 128 Byte Antwort: 128Byte RAM-Inhalt des PICs
3. Set_parameter:
 - 2Byte Anfrage CMD: [0x0AF5]
 - 30 Byte Anfrage: 30 Byte Parameterisierungs Daten
 - 2 Byte Antwort CMD: [0x0AF5]
 - 128 Byte Antwort: 30 Byte Echo der Parametrisierungs Daten + Dummys [0xDD]
4. Set_ckpipe:
 - 2Byte Anfrage CMD: [0x10EF]
 - 30 Byte Anfrage: Dummys [0xDD]
 - 2 Byte Antwort CMD: [0x10EF]
 - 128 Byte Antwort: Dummys[0xDD] oder NAK/ACK Meldung des PICs bei fehlerhaften unvollständigen Parametern (noch zu Klären!)