PICADC3 - simple data acquisition system based on PIC16C774

Wojciech M. Zabolotny

14th March 2005

1 General description

The system allows sampling of maximally 9 analog channels with 12-bit resolution and the sampling rate depending on both: PIC16C774 maximum sampling rate and on the throughput of the communication serial link.

The protocol used in the system is not optimized for the best utilization of link bandwidth it uses some redundant information targeted at easy separation of records and detection of transmission errors.

The current version may work with 115200 or 230400 baud rate. The transmission rate is programmed in the firmware, however it should be easy to modify the firmware so that the transmission speed would be selectable with the jumper.

The system uses “old fashioned” asynchronous serial port. This limits the throughput of the link, but makes possible to implement optoisolation, which is very important in biomedical applications.

2 Communication protocol

After switching on, or after initialization the converter sends the following “welcome message”:

1. Initialization marker RSR_INIT (0xc1)
2. Name of the device “PICADC version_number\n” (eg. “PICADC 3.0.8\n”)
3. Base frequency when using the main xtal (eg. “4608000\n”)
4. Base frequency when using the auxiliary xtal (eg. “16000000\n”)
5. Maximum amount of analog input channels (eg. “9\n”)
6. Maximum amount of sampled channels (eg. “16\n”)

After the above message is sent, the converter is ready for setting of sampling parameters. All values should be set. No defaults are assumed.

2.1 Setting of sampling frequency (“F” command)

This command sends a 21-bit word which defines the sampling frequency in the following way:

- Bits [0..15] - frequency divider
- Bits [17..16] - prescaler setting:
  - “00” - no prescaler
  - “01” - division by 2
  - “10” - division by 4
  - “11” - division by 8
- Bits [19..18] - selection of oscillator:
  - “00” - main oscillator
  - “11” - auxiliary oscillator
  - other values are not allowed

When using the auxiliary oscillator, one must observe limitations regarding the prescaller settings

After reception of each byte, the converter sends it back to the PC. Additionally the checksum of the whole command (without the “F” byte) is calculated and sent back to the PC.

2.2 Setting of sampled and transmitted channels (“C” command)

The converter does not feature the simultaneous sampling of all inputs. Therefore the effective sampling rate for particular channel is equal to:

$f_{smpl} = \frac{f_{base}}{div \cdot N_{smp}}$, where f_{base} is the base frequency, div is the divisor (including the prescaler) and N_{smp} is the amount of sampled channels. This fact leads to some limitations regarding to the effective sampling frequency. For example with base frequency equal to 16MHz it would be impossible to obtain the sampling rate of 1kHz for 7 channels. However it is possible to obtain such sampling rate for 8 channels (with div=2000). Because the transmision link throughput is one of the factors limiting the sampling rate, it is reasonable to add a “fake” sampled channel to obtain the accurate sampling rate, but do not transmit the data received from such channel.

The above model is implemented in PICADC3 by defining the list of sampled channels, and the amount of transmitted channels (starting from the begining of the defined list). The syntax of the commands is as follows:

1. Byte “C”
2. Amount of transmitted channels
3. List of sampled channels, finished with “0x7f” byte.

After reception of each byte the converter sends it back to PC, and additionally after the “0x7f” byte the checksum of the whole command (without the “C” byte, but including the “0x7f” byte) is sent.

The number of each channel in the list may be increased by 0x10 (16) to denote, that the digital inputs should be sampled together with the analog input. The exact moment of digital inputs sampling depends on the software and is not well defined.

Examples:

1. “C 0x04 0x01 0x05 0x13 0x02 0x7f” - this command causes sampling of 4 channels: 1, 5, 3 and 2. The digital inputs are read and sent together with analog value measured in channel 3
2. “C 0x08 0x00 0x02 0x00 0x03 0x00 0x04 0x00 0x05 0x7f” - this command causes sampling of channels 0, 2, 3, 4 and 5. The sampling rate in channel 0 is for times higher than in others.
3. “C 0x05 0x00 0x01 0x02 0x03 0x04 0x00 0x00 0x00 0x7f” - the channels 0, 1, 2, 3, and 4 are sampled. The list is 8 channels long to obtain the accurate sampling rate. Only the first 5 sampled values are transmitted. The values obtained during the last 3 samplings of channel 0 are discarded.

2.3 Start of sampling (“P” command)

After reception of “P” byte, the converter sends it back to the PC, and starts sampling.

The results of sampling are sent as consecutive “records” described in the next section. When sampling is on, the converter accepts only selected commands: the “stop sampling” (S command) , setting of the digital output and the “reset” (0xff command).

2.4 Stop sampling (“S” command)

When the sampling is on, reception of the “S” byte causes end of sampling after the current record (cycle of sampling) is finished. As a confirmation, the “0xc0” byte is sent after the last record. (No copy of “S” byte is sent back, as it could corrupt the data stream).

2.5 Setting of the digital outputs

When PICADC3 receives a byte with MSB set to 1 and when it is not a “Reset converter” command (described below) it treats it as a command to set the digital outputs. The byte (after XORing with 0xff) is written directly to the PORT B, so location of particular bits is a little messed:

bit	7	6	5	4	3	2	1	0
value	1	~DOut3	~DOut2	~DOut1	-	-	~DOut0	-

The bits 3,2 and 0 are not used, however to make sure, that the command does not reset the converter, it is better to set at least one of them to ’0’.

2.6 Reset converter (0xff command)

Reception of the 0xff byte causes immediate hardware reset of the converter. It is obtained by driving low the MCLR pin in the PIC16C774 chip.

2.7 Records with sampled data

The data received by the converter are sent as records with the following structure:

Meaning			B7	B6	B5	B4	B3	B2	B1	B0
Record header			1	0	record number (modulo 64)					
input values from single channel	analog value		0	0	bits 11 to 6 of analog value					
	analog value cont.		0	0	bits 5 to 0 of analog value					
	optional	digital I/O	0	1	bits 11 to 6 of digital I/O					
		digital I/O cont.	0	1	bits of 5 to 0 of digital I/O					
input values from next channels ... (as above)										
Checksum of the record (without header)			0	7-bit checksum						

This organization of data does not assure the best bandwidth utilization, but provides efficient detection of record frames even if the data are slightly corrupted.