

ALGOPIC Manual

author: Winfried Ritsch, Atelier Algorithemics,
ALGOFET Version: 2.0, last updated 31.3.2007

This manual is a short description of the ALGOFET board v2.0 and is published under <http://algo.mur.at/>. Since the ALGOPIC was developed for the „Autoklavierspieler“, many Aspects of the board are mainly tested for this application. Please feel free to report other tests and send me correction etc.

Introduction

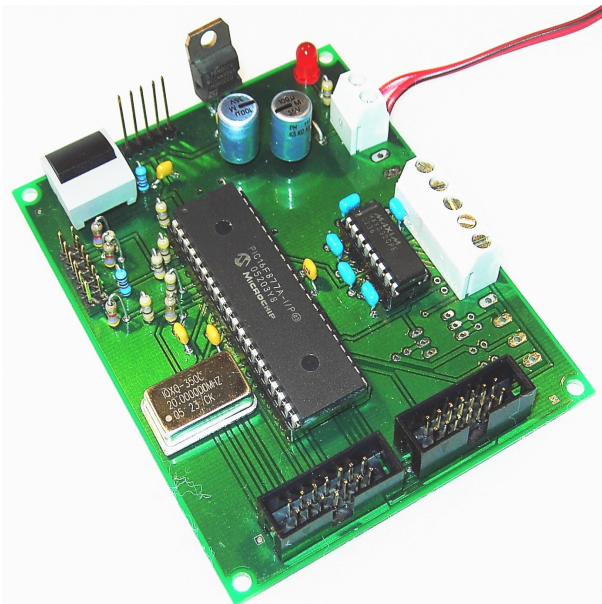


figure 1: ALGOPIC with serial interface

design goals and usages of the board as examples. The board is highly configurable and can be used for many purposes.

The ALGOPIC is a simple micro-controller board for the use of *PIC16F877(A)* or any other PIN-compatible PIC with 40 pins like *PIC18F452*. The purpose of the board is

to interface different hardware boards like ALGOFET and Sensors with an easy to set up configuration.

The board can be interfaced by serial or MIDI Interface from a host and programmed with the ICSP from the ICD2 or any other debugger and programmer, including boot-loader.

It has a jumper-board for configuration like addressing or testing hardware and two IDC ribbon-connectors each 14-pins. Here I will describe the main

PIC Controller

The PIC Controller is mounted on a 40-PIN socket so it is easy to solder. The board is targeted to *PIC16F877 (Version A)* but also a *PIC18F452* can be used. In figure 2 the pin assignment is shown. The Reset is connected to button, which has a lot pads to be soldered at with different kind of push-buttons and also is used by the ICSP¹. RA0-4 is used for the on board Jumper-board or testing Interface. The serial PINS RC6, RC7 are used for either the RS232 or MIDI Interface. All other Pins are used for the external connectors SV1, SV2.

The PIC can be clocked with a Oscillator, but also the internal RC can be used. I recommend the oscillator since seems to be the most stable solution. The frequency can be chosen from recommendation of

1 In-Circuit Serial Programming

the PIC Chip.

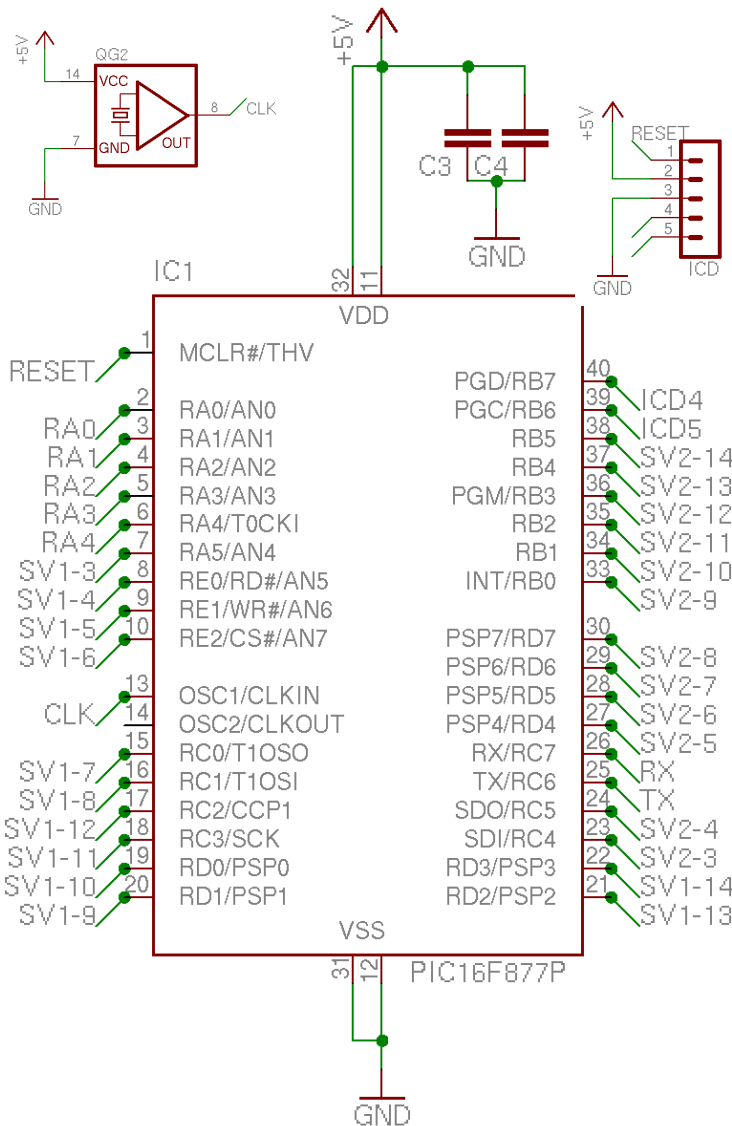


figure 2: PIC pin assignment of a PIC16F877

The ISCP connector

On the ISCP-Connector a Debugger or Programmer can be attached for in-circuit programming or debugging the PIC, see Programmer for more Information. Programming can also be done over a boot-loader (see software description).

The pin assignment is made so it is compatible with the PICStart+² and compatible Kits.

Pin	ICSP
1	Reset
2	+5V
3	ground
4	PGD of PIC
5	PGC of PIC

Power Supply

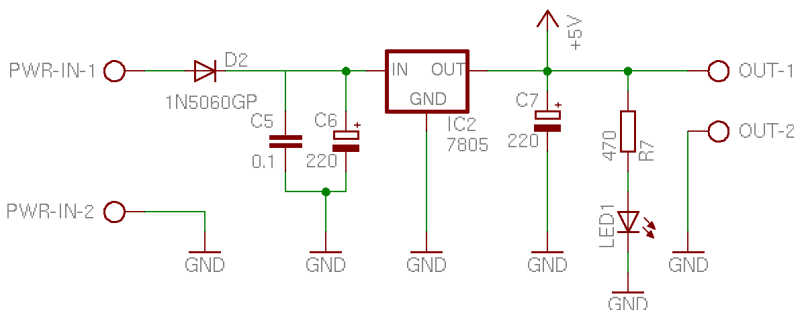


figure 3: Power circuitry

As a power supply 7-12V can be used, which are regulated by a 7805 voltage stabiliser to supply the PIC and external circuitry with current. 200mA is needed for the PIC and oscillator without external power needs.

and a stabilized voltage of 3-5V can be applied direct to the PIC. If more boards are used the OUT-1 and OUT-2 clamps can be used to supply more boards with only one stabilizer or any other circuit which needs 5V. The diode D2 is to protect

Also this can be omitted

against wrong polarity on input, but also can be bridged, so the input voltage can be 0.6V lower.

Serial I/O

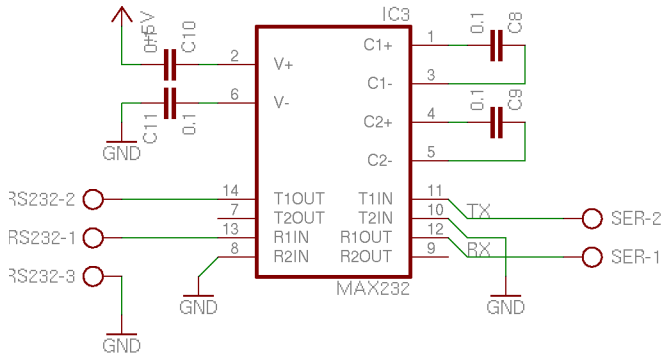


figure 4: serial circuit

The serial I/O can be used with RS232 Standard or MIDI. Only one of these options is possible at the same time.

If RS232 is used a MAX232 is needed to convert RS levels (+/- 12v) to TTL-levels. Also the outputs of the MAX232 can be used to be connect over clamps another ALGOPIC RX-PIN, so a daisy-chain of more ALGOPICS can be build, to control these with one host. The ALGOPIC can be directly driven by TTL-levels at these Ports SER-2 for TX and SER-1 for RX.

MIDI³ I/O

If the MIDI circuitry is used, MIDI-IN is connected over an optoelectronic coupler like CNY17 and the output directly driven by the TX-PIN of the PIC which delivers enough power for the 5mA current loop technology of MIDI.

No MIDI-Through is implemented since it would need a buffer IC and can also be done with merging in software to build a daisy-chain network of more MIDI devices.

For the right bit-rate carefully choose the PIC Oscillator frequency.

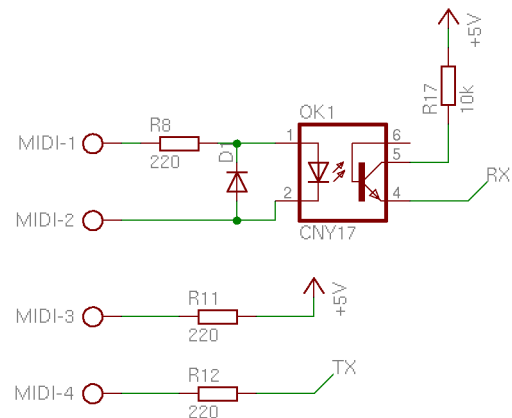


figure 5: MIDI Port

jumper board

The jumper board can be used for different purposes dependent on the software used. The original usage is to set the operation mode and addresses used by the board, if more than one is on the same serial control to be identified.

So Prog/Addr, Addr1, Addr2, Addr3 can be used for that. You can directly plug jumper on it. Jumper set is TTL zero.

Additionally these boards can also be used as analog-Converter inputs, if configured by the software. So you can attach either sensors or switches to that. Like shown in figure 6 there is also a JP2-5 prepared to connect a potentiometer directly. Therefore also 5V and ground is supplied on two extra pins JP2-6

As an example I build a testing circuitry with two buttons and one potentiometer to control the ALGOFET board without a host see figure 7.

3 Music Instrument Device Interface, a Specification of MMA, free MIDI Specs can be found under <http://www.borg.com/~jglatt/tech/midispec.htm>

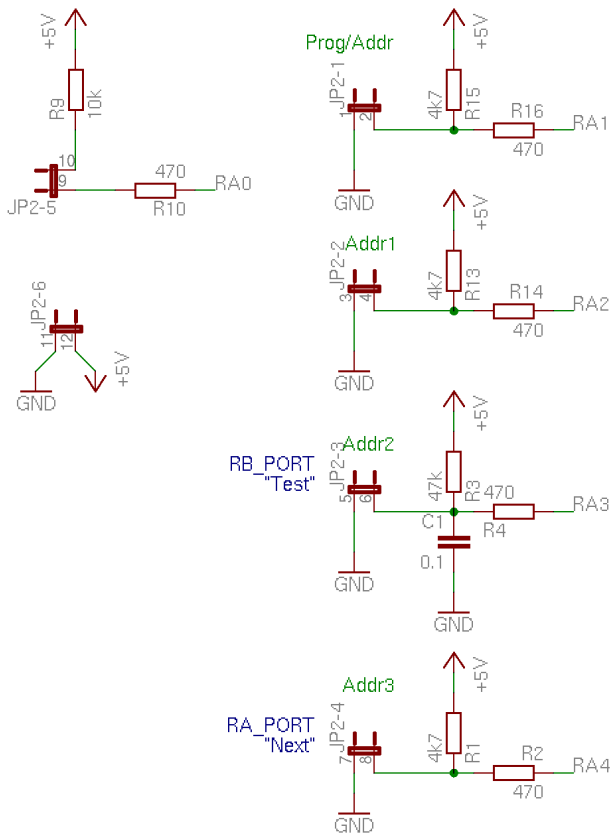


figure 6: jumper board
External connector



figure7: direct input hardware

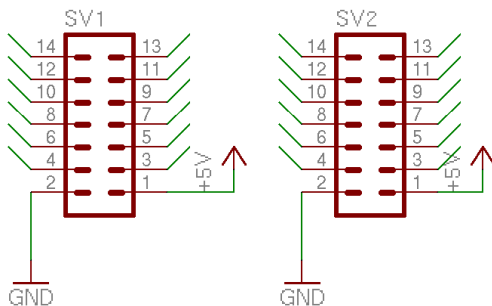


Abbildung 8: external connectors for ribbon cable connector

There are two external connectors to connect to other hardware like the ALGOFET. For this multipoint socket connectors are in conjunction with ribbon cables can be used. On these there are 12 pins from the PIC are connected and +5V and ground. So also the secondary circuit can be powered over this connector.

No pull-up resistors or so are used, so that each pin can be used as in- or output. See the PIC specifications for more details on connection

SV1				SV2			
Pin	description	Pin	description	Pin		Pin	
1	+5V	2	ground	1	+5V	2	ground
3	RA5/AN4	4	RE0/RD#/AN5	3	RC4/SDI	4	RC5/SDO
5	RE1/RW#/AN6	6	RE2/CS#/AN7	5	RD4/PSP4	6	RD5/PSP5
7	RC0/T1OSO	8	RC/T1OSI	7	RD6/PSP6	8	RD7/PSP7
9	RD1/PSP1	10	RD0/PSP0	9	RB0/INT	10	RB1
11	RC3/SCK	12	RC2/CCP1	11	RB2	12	RB3/PGM
13	RD2/PSP2	14	RD3/PSP2	13	RB4	14	RB5

Software

As an example software a the control of the ALGOFET board with 12 solenoids is shown, called "spieler". The main focus is the serial Interface and Pin assignment for this application.

Since timing is critical the use of an assembler is needed. As development tool the open source free IDE piklab⁴ is used now, the first applications has been developed with MPLAB⁵. As assembler the giasm from gputils⁶ are used.

As a programmer and debugger ICD2⁷ was used, but any other programmer or debugger can be used for this, which supports ICSP.

How spieler works

The specifications for the player was from an application , where eight ALGOPICS control 16 ALGOFET boards and with them 88 stroke magnets for a playing a piano with 88 keys. But also can play up to 96 any other electrical devices with DC:

- parallel ALGOPICS on one serial line
- serial input with 115200 kBaud
- Timing down to accuracy of 400us of pulses
- 128 different stroke pulses and arbitrary hold time
- Testing hardware for playing without serial host
- security so solenoids does not burn on host failure

The syntax for the serial board see the ALGOFET Manual

Program layout

The critical parts are run in the interrupt routines, where a timer interrupt controls the finest time grain t_{int} and the serial interrupt is dispatched to a parser shown in figure 9. With the first counter all timings can be scaled with the scaling factor K to get the time ticktime= $K*t_{int}$. This ticktime is the smallest unit to play a pulse as attack. The effective pulls width is scaled by the velocity, a seven bit data (0-127). With the factor N the time is scaled to get a long period in which the activity of the host is

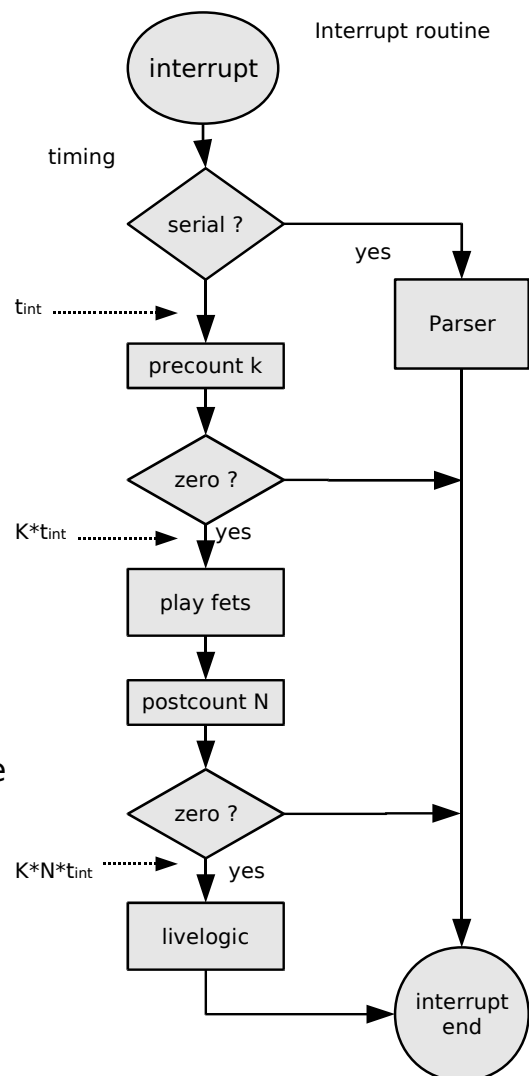


figure 9: interrupt routine

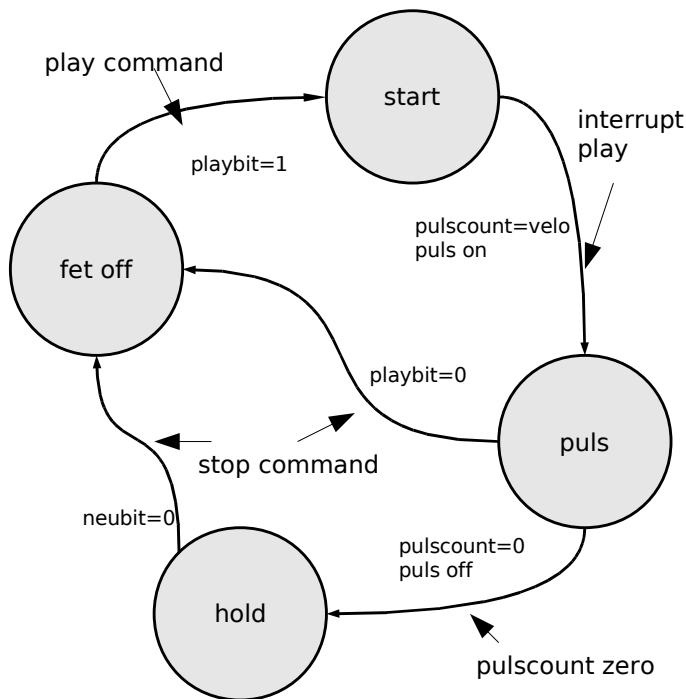
4 Piklab is an integrated development environment for applications based on Microchip PIC and dsPIC microcontrollers similar to the MPLAB environment (see <http://piklab.sourceforge.net/>).

5 IDE from microchip, see <http://www.microchip.com/>

6 a collection of tools for the Microchip (TM) PIC microcontrollers <http://gputils.sourceforge.net/>

7 In Circuit Debugger 2, from Microchip (TM), also available as third party product.

checked and if no signal has been sent in between two zeros all FETs will be turned off.



The control of each output (FET) is done with the state machine shown in figure 10. Since the program is small, we use the variable assignments are all in *BANK 0*. Also we hold all code, especially the interrupt code in one page for short jumps to be fast.

On a 20MHz PIC we can do 5000000 instructions. Since of the smallest timetick of 400us, we can use up to 2000 single instruction in one loop, without getting something lost.

For 12 outputs it means (2000-instructions of parser - overhead)/12 which is at least 150 instructions which can be done easily

figure 10: state machine for each FET

Serial parser

The parser works a little bit like a MIDI-Parser, using bit 7 as status-bit of each 8Bit values.

The trick is to use the Bit5-7 of the status bits as address, which are compared with the jumper setting of JP2-2 to JP2-4, read at start-up in the initialization phase. So up to 8 ALGOPICs can be addressed on the same serial line, therefore the daisy-chaining of serial-out to serial in can be used. bit 0-4 of the status byte is used to drive 16 commands to which the following received data-byte is associated.

For playing the 12 outputs, command 0-11 is used, where the second byte is velocity. Command 12 sets the scaling factor of the system (see above).

A short abstract description of the parser and it variables is done in a readme.txt file in the software directory.

Disclaimer

I cannot be held responsible for the information contained within this docs and hardware documented here, even I tried hard not to make failures but nobody is perfect. Any uses and/or misuse of the information that could result in bodily and/or mental harm to you or any other person that came in contact with the information cannot hold me or any company or anyone else responsible.