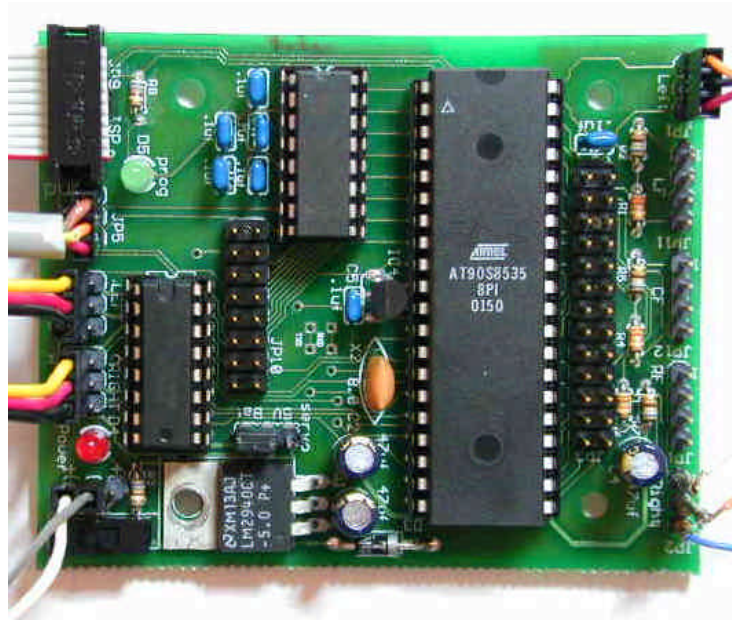


AVR Robot Controller 1.1



Atmel AVR based

- Powerful 8-bit RISC processor, about 10x faster than a PIC or HC11 when coding in C.
- Upgrade CPU's available that double memory, speed and expand I/O possibilities.
- Free high quality tools available: assembly, BASIC, C, and simulators.
- Inexpensive in-circuit debugging tools available.

Complete solution for small robots

- Serial port.
- Dual H-Bridge motor driver.
- Connectors for two additional R/C servo output or encoder inputs.
- Connectors for five analog or digital inputs.

Expandable and flexible

- All processor signals + power brought to expansion headers: 8 analog and 24 digital lines.
- Mounting holes compatible with LEGO DACTA hardware.
- +/- 10v available for low powered analog circuits.
- Expansion headers bring all CPU signals to .1" perf-board compatible pins for experimentation.

Table of contents

Kit contents.....	2
Assembly instructions	3
Hardware Description	6
Programming.....	8
Software Tools.....	9
Schematic	10

Kit contents

Item	Amount	Description	Vendor	Part#
1	1	ARC PCB	Barello.net	ARC 1.0 Board
2	1	ATMEGA16-16PC	Digikey	ATMEGA16-16PC-ND
3	1	ZTT-16.0 Ceramic Resonator	Digikey	X908-ND
4	1	LM2940CT-5.0 TO220 case	Jameco	107182
5	1	Diode, 1A	Digikey	1N4001GCT-ND
6	3	47uf 20v Electrolytic	Digikey	P11196-ND
7	8	.1 uf ceramic Z7T .1"	Digikey	P4924-ND
8	3	33K 1/8 wt carbon film	Digikey	33KEBK-ND
9	5	680 1/8 wt carbon film	Digikey	680EBK-ND
10	3	330 1/8 wt carbon film	Digikey	330EBK-ND
11	2	1k 1/8 wt carbon film	Digikey	1.0KEBK-ND
12	1	LED red	Digikey	160-1078-ND
13	1	LED green	Digikey	160-1080-ND
14	2	16 pin low profile DIP socket	Jameco	112221
15	1	40 pin low profile DIP socket	Jameco	112310
16	1	Switch, SPDT, PCB mount	Digikey	EG1903-ND
17	5	1x3 .1" pin header	Jameco	
18	3	1x4 .1" pin headers	Jameco	
19	2	2x3 .1" pin header	Jameco	
20	2	2x5 .1" pin header	Jameco	
21	2	1x3 .1" pin socket	Jameco	157382
22	6	Crimp sockets	Jameco	100765
23	1	TI 754410 Dual H-Bridge	Digikey	296-9911-5-ND
24	1	DB25 Solder Cup male	Jameco	15114
25	1	DB9 Solder Cup Female	Jameco	15771
26	1	XXX232A RS232 Driver	bgmicro	ICSMAX232A
27	3'	22ga 3 conductor cable	Jameco	31860
28	2	10 pin IDC socket	Jameco	32491
29	3'	10 conductor ribbon cable	Jameco	135538
30	1	.1" shorting jumper	Digikey	S9002-ND

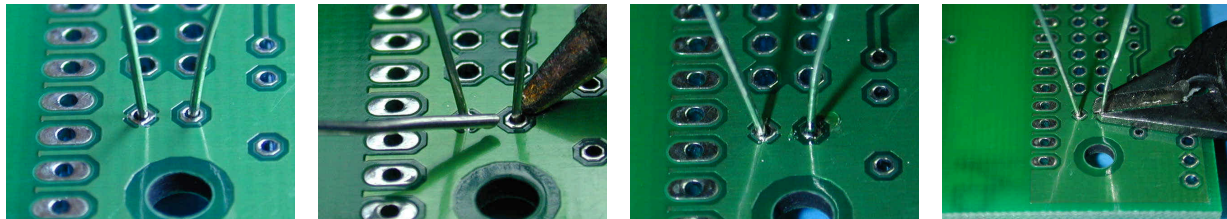
Assembly instructions

Overview

The ARC can be assembled using a fine tip 25-40wt soldering iron, thin 60:40 rosin core solder, Flush side cutters and a pair of small needle nose pliers. Additional items that are useful are a solder tip cleaner (damp sponge or cloth), paste solder flux and a small gauge wire stripper (18-26). Because it is fairly small you need a well-lighted area to work in. A magnifying glass might be useful for inspecting solder joints.

The side cutters are used to trim component leads flush to the board after soldering them. With a little practice the side cutters can be used to strip wire as well: the trick is to cut into the insulation, but not the wire and then pull the end off.

The key to obtaining a good solder joint is to get everything hot enough so that the solder “wets” the surface and wicks into the spaces between the wire and what you are soldering it to. First clean the iron tip by wiping it across a damp sponge. Then tin the tip with a bit of fresh solder. Then place the tip against the pin and the solder pad i.e. wedging it in the corner of the two. Then touch the solder to the pin or the board near to the tip. Usually it takes just a fraction of a second to get the joint hot enough to melt the solder. Then leave the tip on the joint for a couple seconds more until the solder visibly wicks into the hole. If the hole looks empty you can put another touch of solder on. When done, snip the excess lead material off flush to the bottom of the board. Usually cleaning and tinning the tip is needed only between rounds of soldering or after soldering 20-30 joints in a row



You need a workspace as well. The ARC uses static sensitive components and you need to exercise care while assembling it. The best is an anti-static mat with a wrist strap and grounding it to the same ground as the soldering iron uses. An alternative is to work on a bare wooden surface. Wood neither conducts or accumulates static electricity. Sometimes you can obtain a sheet of conductive foam and work on that. Keep it grounded through a large resistor (e.g. 1-2 meg ohms) and make sure some exposed skin touches the surface before you pick up static sensitive parts (anything that looks like an IC). Once the board is assembled and the IC's inserted it is much less sensitive to static discharge.

Prepare boards

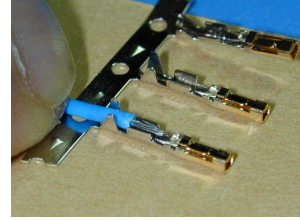
The PCB comes with a serial adaptor and a programming adaptor that need to be detached. First break the long line of holes. You might find it easier to break if you run a razor knife along the top and bottom of the line of holes. Then snap the two smaller boards apart. Finally, if you want, you can sand or file the rough edges smooth.

Serial cable

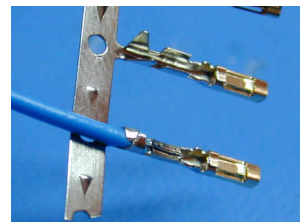
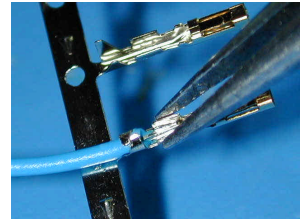
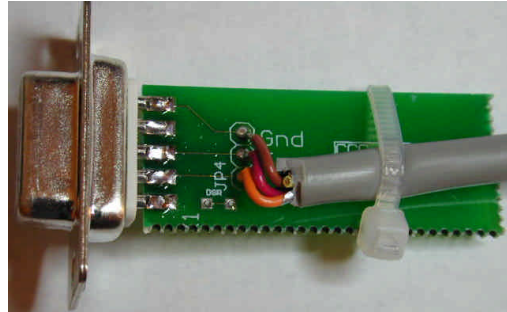
The small PCB, DB9 connector and the 3' gray cable are used to make a serial interface cable that can be used to connect the ARC board to your PC. This is useful when debugging your programs as it allows the program to tell you what is going on.

- ❑ Strip on inch of gray insulation from both ends of the serial cable and expose the wires. Snip off the bare wire and one of the four insulated wires.
- ❑ Strip 1/4" of the ends of each remaining wire and tin the exposed ends.
- ❑ Trim the tinned wire to 1/8" long and solder to the crimp pin sockets as shown below.

- ❑ Crimp the pins to the wire and assemble a 3-pin socket. The pin sockets should snap into place when inserted correctly.
- ❑ Mark 3-pin socket with GND, Tx and Rx as shown in the photo. Each line will have a corresponding label on the PCB.
- ❑ Solder DB9 connector to board, pay attention to the orientation so each solder cup covers a pad.
- ❑ Solder the wires to the board.
- ❑ Apply strain relief. Tape or a nylon cable tie works well.



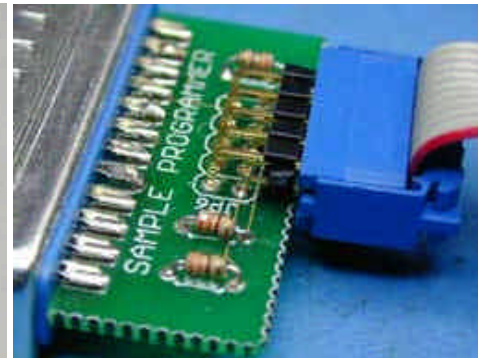
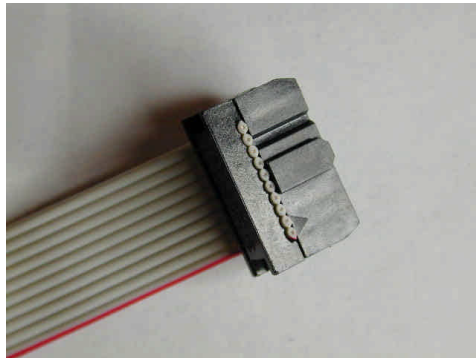
Finished Serial Connector



Programming cable

The kit is supplied with material to make a programming cable that works with the BASCOM Basic Compiler or the avr-gcc C compiler program “avrdude”. BASCOM and Avrdude work with any third party programmer that has an Atmel 10-pin programming socket, so you only need to build the cable if you don’t already own a programmer.

The programmer uses a male DB25 connector to plug directly into a PC printer port. Assemble the board using three 330 ohm resistors, the 2x5 pin *right angle* header and the DB25 connector. The cable is assembled with a 10-pin IDC plug on each end. Assemble the plugs as illustrated below with the red line oriented towards the “pin 1” arrow.



Finished Programming Cable

Controller

Part locations are labeled on the board, but some labels are incomplete due to solder pad location. Please read the instructions to insure that parts are installed correctly.

The control board has been designed with oversized holes to facilitate hand assembly and disassembly. The problem is that components will fall out when the board is turned upside down. If you assemble from smallest to largest components, then it is possible to load several parts onto the board, cover it with a piece of conductive foam or cardboard and turn it over. Still, parts will tilt and move about. The trick is to solder one pin of each part into place, turn the board over and re-position the parts by re-heating the soldered pin while pushing on the other side

with your finger. When all the parts are positioned to your satisfaction, then turn the board back over and finish soldering the rest of the pins.

Solder components

The following list is in order of lowest to highest component. All components in each section can be soldered at one time, or each component can be soldered in separately.

❑ **Resistors & diode.**

The diode is inserted with the band towards the end of the arrow on the board. If you know what you are doing and want maximum power you may omit the diode, replacing it with a short piece of wire. This will remove the reverse battery protection that can destroy the H-Bridge chip and/or attached servos. The voltage regulator protects the rest of the board from battery reversal.

NOTE: the kit is supplied with 680 (blue, gray, brown) and 33k (orange, orange, orange) resistors for use in the analog section. These resistors are selected to work well with the QRD1114/3 series photo reflex detectors. *If you don't plan on using these detectors don't solder these parts in.* Other photo reflex detectors require differing values. For example the QRB1134 works better with 180 & 4.7k resistors, respectively. Use appropriate substitutes, or read the section titled "Left, Center & Right floor sensors", below.

❑ **Small capacitors, LED's & switch**

The Red LED is power and the green LED is program. The LED's have a flat side that is oriented towards the near edge of the board where they are inserted. There is a long and a short lead; the short lead should be near the edge of the board and the longer lead towards the center of the board. Orientation of the capacitors and switch is not important.

❑ **IC Sockets & voltage regulator.**

All IC's can be mounted directly on the board. However, if you are an experimenter you probably should use the IC sockets so the IC's are easy to replace. Make sure the notch on the bottom of the socket matches the notch in the markings on the board.

If you want to maximize the power handling capabilities of the 754410 H-Bridge, you must solder that chip directly onto the board. The ground plane of the board provides the heat sink needed for maximum power output. This is not recommended for beginners.

Bend the leads of the voltage regulator ½ way through the fat part, near the body of the IC. This will align the mounting hole over the hole in the PCB and the IC should lay flat when soldered in.

❑ **Crystal & reset chip**

Install to match the legend on the board. The *reset chip is not installed* when using an Atmega16 CPU as it has the reset circuit built in. The internal reset circuit needs to be enabled when programming the chip. Please see

PROGRAMMING, below for more details. The crystal is optional too. The mega16 has an internal 1, 2, 4 or 8 MHz oscillator. In addition, the mega16 can operate with an external crystal of up to 16mhz. .

❑ **Headers.**

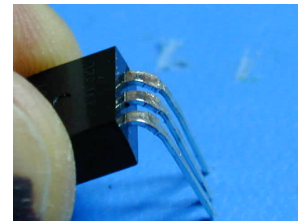
Which headers are installed has to be decided upon before soldering them in place. The three and four pin headers are used for power, signaling and serial communications. You need to decide whether to use the pins & sockets or directly connect wires. The kit comes with two 3-pin sockets. One is for the serial line and the other is to fabricate a power connector. However, if you have your own power disconnect, you might find it easier to directly solder the power wires into the board.

Some sensors, like the Sharp GP2D12, come with a socket and plug. The wires on the plug can be soldered directly to the board. Other sensors that have no disconnect might better be attached via sockets and headers.

The 26 and 16 pin headers (for board expansion) are not supplied with the kit.

❑ **Large capacitors**

Pay attention to the orientation of these devices. The longer lead goes into the hole marked with a '+'. The gold bar on the side of the capacitor should be on the side marked with a '-'. sign.



Cleaning and Inspection

After all components are installed inspect the bottom of the board with a magnifying glass. Look at each solder pad and make sure that the solder is shiny and has a “wet” look. Reheating and applying a touch of solder can rework pads that are incompletely filled with solder or look gray and dull.

Look for solder bridges between pads or pins that are close together. If you used a fine tip iron and no more solder as needed for a good joint, you shouldn't have any bridges.

You can clean the flux (from the solder) from the board with denatured ethanol (alcohol). Use an old toothbrush to scrub all the pads with the alcohol

After inspection apply power to the board. Power can be any source from 6-24 volts, DC. Pay attention to the polarity symbols near the power plug. With the switch in the on position (towards the near edge of the board) the red power LED should light.

Install chips & test board

Now you can install the various chips. All chips are installed with the notch (pin 1) oriented away from the switch and voltage regulator. See the section on programming (below) for testing out basic functions of the board.

Hardware Description

The ARC is based upon an Atmel AT90S8535 microprocessor. This processor has 8 Kbytes of code space and 512 bytes of RAM. The Atmega163 chip is pin and code compatible and doubles the code and ram space. The Atmega16 chip doubles the clock speed of the Atmega163. The ARC 1.1 ships with the Atmega16 chip pre-programmed with the BASCOM sample “FLASHLED.bas”. In the near future the ATMEGA8535 will be available. All chips will function in the board with minimal software changes, the primary difference being the amount of internal memory available.

The CPU has four 8-bit I/O ports and many peripheral functions that work through these I/O pins. Port-A also doubles as the Analog input port. Port D doubles for many of the built in functions like serial I/O, PWM and timers. Port B doubles as the programming interface and Port C is just plain vanilla. Below is a map of all I/O functions that are brought out to headers on the ARC. The expansion headers are not included since they bring out all I/O port bits as well as power and ground.

Function	CPU I/O Port and Bit	Function	CPU I/O Port and Bit
Left Analog input	PORT A, Pin 0 (ADC0)	Left Motor Direction	PORT C, Pin 3
Left Floor Analog input	PORT A, Pin 1 (ADC1)	Left Motor PWM	PORT D, Pin 4 (OC1B)
Center Floor Analog input	PORT A, Pin 2 (ADC2)	Left Motor Channel A	PORT D, Pin 3 (INT1)
Right Floor Analog input	PORT A, Pin 3 (ADC3)	Left Motor Channel B	PORT C, Pin 2
Right Analog input	PORT A, Pin 7 (ADC7)	Left Servo Output	PORT C, Pin 2
Right Motor Direction	PORT C, Pin 4	Option A	PORT B, Pin 6 (MISO)
Right Motor PWM	PORT D, Pin 5 (OC1A)	Option B	PORT B, Pin 7 (SCK)
Right Motor Channel A	PORT D, Pin 2 (INT0)	Option C	PORT B, Pin 4 (SS)
Right Motor Channel B	PORT C, Pin 5	Prog LED	PORT B, Pin 4 (SS)
Right Servo Output	PORT C, Pin 5	Serial Input (Rx Data)	PORT D, Pin 0
		Serial Output (Tx Data)	PORT D, Pin 1

Power

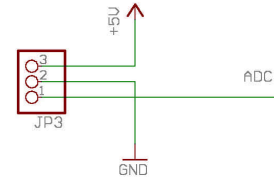
Three pin header with center ground. The ARC board is protected against battery reversal with a diode. This diode reduces the available power to the board and prevents destruction of the H-Bridge in case of accidental battery reversal. If you want, you can use a polarized battery connector and solder it in directly to the board and replace the diode with a bare wire.

Expansion headers JP3 and JP10

Refer to the schematic for the connections to these headers. All CPU I/O, +5v, ground, +/- 10v and battery voltages are supplied. JP3 supplies +5v and ground from the analog section. JP10 supplies +5v, ground, +10v and -10v from the digital section. The +/-10v is from the serial interface chip and can only supply about 10ma of current. This is sufficient for one or two low powered op-amps. JP3 and JP10 are oriented with pin 1 towards the notch side of the CPU.

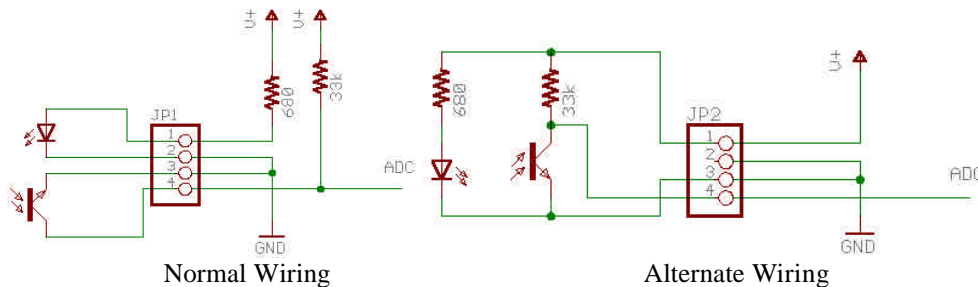
Left/Right analog input

These are raw analog inputs with filtered +5v and ground. They are intended to directly connect to the Sharp GP2D12 or 15 sensors. However any 0-5v output sensor may be used. You can also connect a potentiometer to these inputs for position feedback or variable input. If using the Sharp sensors, you can solder their wires directly into the board and save space and headers. The sensors have connectors at the sensor that can be used to detach the sensor from the board.



Left, Center & Right floor sensor inputs

These connectors have four pins intended to match the pin-out of QRB1134 photo-reflex sensors. The resistors supplied with the kit are selected to work well with these sensors. The sensors come with 2' of attached wire. Pin 1 is attached to the outer E (emitter) wire. Pin 4 is attached to the outer S (sensor) wire. If you do not want to commit the board to these sensors, you can replace the 680-ohm resistor with a piece of wire and leave out the 33k resistors. The resulting circuit is similar to the Left and Right analog inputs and allows +5v-powered sensors to be connected to the board.



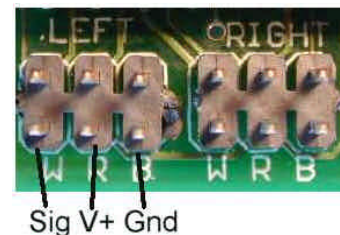
Left/Right motor connectors

The left and right motor connectors provide output from the H-Bridge, power, ground and two CPU I/O lines. The additional lines can be used to implement an encoder feedback or input mechanism, or, can be used as output lines to drive a standard R/C servo.

R/C servo output

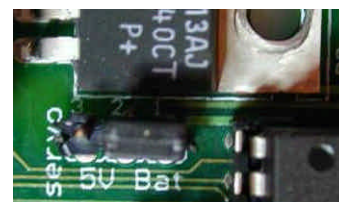
When driving an R/C servo, use the pins nearest to the edge of the board. They are marked with letters indicating polarity: B, R & W for Black, Red and White. Some servo brands substitute yellow for white.

When driving servos the board has the option of selecting regulated +5v or raw battery voltage to drive the servos. R/C servos are usually rated at 4.8v and 6.0v, but can be driven with 9-12 volts without damaging the internal electronics. Higher voltages will increase speed and torque of the motors. A rough rule of thumb is that doubling the voltage (9.6 vs. 4.8) doubles the RPM and the maximum torque of the servo.



Servo Power

A 3-pin header used to select the power source for the Motor Headers when used for R/C servos or encoders. There are two options: regulated +5 volts, or the battery supply. The H-Bridge is always connected to the main power source.



H-Bridge output

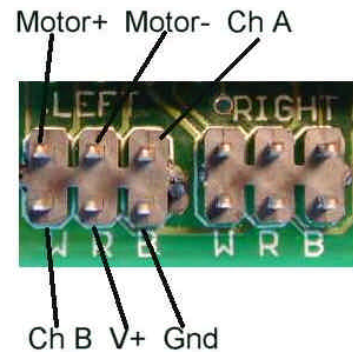
The H-Bridge output is on the inner pins of the header and is directly connected to the 754410 H-bridge. This can supply up to 1a with a battery voltage of up to 24v. If you gut a servo and re-attach the servo cable wires to the motor, be aware that the two h-bridge output pins are not aligned with the power pins: you either need to wire the servo motor to Red, White, or connect the plug in reverse.

Encoder input

When driving motors with the H-Bridge, the additional two signal wires and power can be used to gather encoder feedback. When driving an encoder make sure the jumper block labeled **SERVO** is connected for +5v.

	Left Motor	Right Motor
DIR	Portc.3	Portc.4
PWM	Portd.4 (OC1B)	Portd.5 (OC1A)
CH-A	Pind.3 (INT1)	Pind.2 (INT0)
CH-B	Pinc.2	Pinc.5

Motor Port I/O assignments



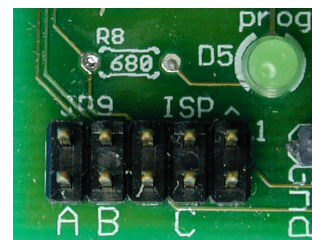
ISP

The 2x5 pin header labeled ISP is for In System Programming of the CPU. This is the mechanism by which the CPU is loaded with a new program. The header has a dual use. When the programming cable is removed, several of the pins may be used to select options or give feedback. All pins are available from the expansion headers – refer to the schematic for more details. In addition one pin is used to control the program LED (green). This gives both visual feedback while programming, but can be controlled by the users program to give feedback.

Option select input

When a shorting bar is across the inputs, the corresponding pin value will be zero. There are no pull-up resistors on these inputs, so you need to enable the CPU pull-up on those ports if you intend to use them as inputs. Reset has no CPU readable input. It simply resets the CPU when shorted.

Pin	Pair label	Direction	IO port
9	A	Input	Pinb.6 (MISO)
7	B	Input	Pinb.7 (SCK)
5		Input	CPU reset
3	C (Prog LED)	Input/Output	Pinb.4/Portb.4
1		Output	Portb.5 (MOSI)



Options I/O port assignments

Program LED output

The program LED output is inverted. I.e. writing a 0 to portb.4 will illuminate the LED, writing a 1 will turn it off. Remember to configure that I/O pin as an output or it will never turn on. When using Option C as an input, you cannot use the LED (they are connected together).

Programming

The In-System-Programming connector works with any Kanda compatible 10-pin AVR programmer. The cable connects to the ISP header with the pin 1 marker (a triangle) next to the 1 and caret printed on the board.

The ATMEGA16 CPU's, as provided in the kit have been programmed to work with the internal 8 MHz oscillator and have the internal reset circuit enabled. If you want to use the external 16 MHz crystal, you need to modify the

CPU fuse settings. The fuse settings may be modified with built in programmer that comes with BASCOM. It is necessary to refer to the chip data sheet (www.atmel.com) to determine precisely which bits need to be modified.

The cable provided with the ARC works with the SAMPLE programmer provided with BASCOM or with the Winavr (gcc) C supplied “avrdude” programmer. While power is off, plug the cable into the board and the Line Printer port of a PC. Load the sample program “FLASHLED” from the ARC web site, power up the ARC and click on the Compile button, then click on the Run button. The **Sample** programmer should pop up with the ARC CPU identity listed (m16). Click on the Auto-Program button and the program should load in a couple seconds. The green **program** LED flashes with a one second cycle time after a successful download. If the LED flashes very slowly, then the oscillator options for the CPU have not been set and the CPU is likely running off the factory default 1 MHz internal oscillator. You can re-program the chip to run off the external crystal/resonator, or the internal 8 MHz oscillator by reading about the fuse bits and programming them appropriately.

If you have an STK200/300 compatible programming cable you can use that with BASCOM by selecting the STK200/300 programmer in the Options->Programmer menu. Please refer to the BASCOM documentation for using other programmers (all are supported).

Software Tools

Free, high quality tools are available for the Atmel AVR series of microprocessors. You can download tools from the following sites:

www.atmel.com/atmel/products/prod23.htm

Assembler, Simulator, Application notes, software

www.avrfreaks.net

Discussion groups, sample code & projects, AVR-GCC C compiler

www.mcselec.com

The BASCOM compiler and various application notes can be found here.

www.barello.net/ARC

AVR Robot Controller resource page, user guide and sample code

<http://groups.yahoo.com/group/AVRRobotControl>

Technical support, applications, code fragments, documentation, design files.

Schematic

