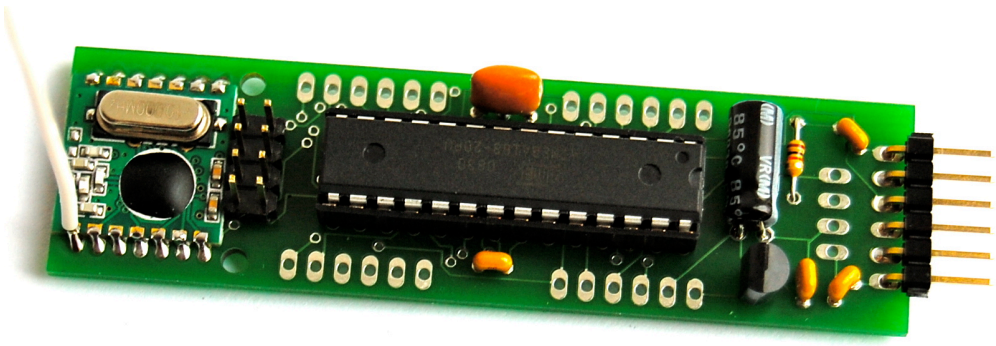
JeeNode V1

*A small *duino-ish MPU board with a wireless RF module*

Jean-Claude Wippler
jeelab.equi4.com
February 2009

Overview

The "JeeNode" is a small micro-controller board which can be used for a variety of Physical Computing[1] tasks. From measuring things like temperature and other environmental data to tracking / controlling energy consumption around the house. It was inspired by the fascinating Arduino[2] board and its IDE, which open up a whole new world of C/C++ computing on tiny devices.



What's on board

- an ATmega168 or ATmega328 MPU, by Atmel
- a RFM12B wireless RF module for the 433, 868, or 915 MHz ISM band, by Hope RF
- a 3.3V power regulator which accepts 3.5 ... 12V as external power source
- four I/O ports, each with 1 analog I/O, 1 digital I/O, +3.3V, ground, PWR, and IRQ line
- a 6-pin FTDI-compatible serial I/O port
- a 4-pin combined TWI/Power connector
- a 2x4-pin combined ISP/SPI connector, with 2 select lines

[1] http://en.wikipedia.org/wiki/Physical_computing
[2] http://arduino.cc/

Similarities with the Arduino boards

- they run the same bootstrap code, and can therefore run Arduino IDE “sketches”
- same micro-controller as the current range of Arduino / Freeduino / and other *duinos

Differences with the Arduino boards

- there’s a wireless RF module on board, whee!
- completely different physical layout, incompatible with Arduino “shields”
- a JeeNode runs internally at 3.3V, not 5V - this also affects all I/O ports & pins
- not all the Arduino pins are brought out on connectors, pins are assigned fixed roles
- the SPI connector has an extra 2 pins, to allow re-using it as SPI bus for 1 or 2 devices
- the TWI bus has been permanently assigned to its own connector
- runs at 16 MHz - but with a less accurate ceramic resonator i.s.o. a crystal
- there are no LEDs, to reduce power consumption when running off batteries

Rationale

The main reason for creating the JeeNode was that Arduinos are not yet very convenient for 3.3V devices such as the RFM12B module. By including a 3.3V regulator, the JeeNode can be powered from USB with an FTDI cable/board, a DC power adapter, or various types of 3.6 ... 4.5V batteries.

The second reason was to have a configuration which includes wireless connectivity by default. The RFM12B module is a low cost option with sufficient power and range to provide reliable communication around the house - a basic packet protocol can be implemented in just 2 Kb of C code.

The third reason is that I constantly ran out of power and ground connectors when hooking up a few sensors to an Arduino. While there are nice “proto shields” to overcome this, it seemed logical to try and come up with a different connector scheme for the common case of just a few sensors / actuators. Furthermore, being identical, all four JeeNode ports have the same features and connections, allowing sensors to be switched around as needed.

Supporting software

JeeNodes are so similar to Arduinos that they can be used with the same IDE as Arduinos. This was a major design choice. Not only is the Arduino IDE very convenient with so many libraries written for it, it also happens to run equally well on Windows, Macintosh, and Linux.

The following libraries have been added to better support JeeNodes:

- The "Ports" library makes it easy to connect sensors to a port, and to re-use the same code when re-connecting those sensors to a different port later on. It contains a few extras over what the standard Arduino / Wiring code offers: a bit-banging TWI driver which can run on any port (even on all of them in parallel) and simple shift functions to drive SPI-like devices.
- The "RF12" library contains a driver for the RFM12B module. It makes it easy to send packets of up to 66 bytes of data around, either as broadcasts or in point-to-point mode. There is some logic to support sending acknowledgements, as starting point for building more reliable communication on - with retransmits and timeouts to deal with packet errors and loss.

These libraries, and all other code related to the Jee projects can be found at Google Code[3].

[3] http://code.google.com/p/jeelab/source/browse/trunk

Pinout - JeeNode V1

(pins are numbered 1...up from left to right when looking towards the upper side of the board)

Ports 1 ... 4

Pin	Name	Description
1	IRQ	interrupt (tied to INT1)
2	DIO	digital I/O line
3	GND	ground
4	AIO	analog I/O line
5	VCC	+3.3V
6	PWR	external power (+)

FTDI connector

Pin	Name	Description
1	GND	ground
2	N.C.	not connected
3	PWR	external power (+)
4	RX	serial receive
5	TX	serial transmit
6	DTR	for bootstrap / reset

ISP/SPI connector

Pin	Name	Description
1	SEL0	tied to PB0 (A. pin 7)
2	SEL1	tied to PB1 (A. pin 8)
3	MISO	master in / slave out
4	VCC	+3.3V
5	SCK	SPI clock
6	MOSI	master out / slave in
7	RST	reset
8	GND	ground

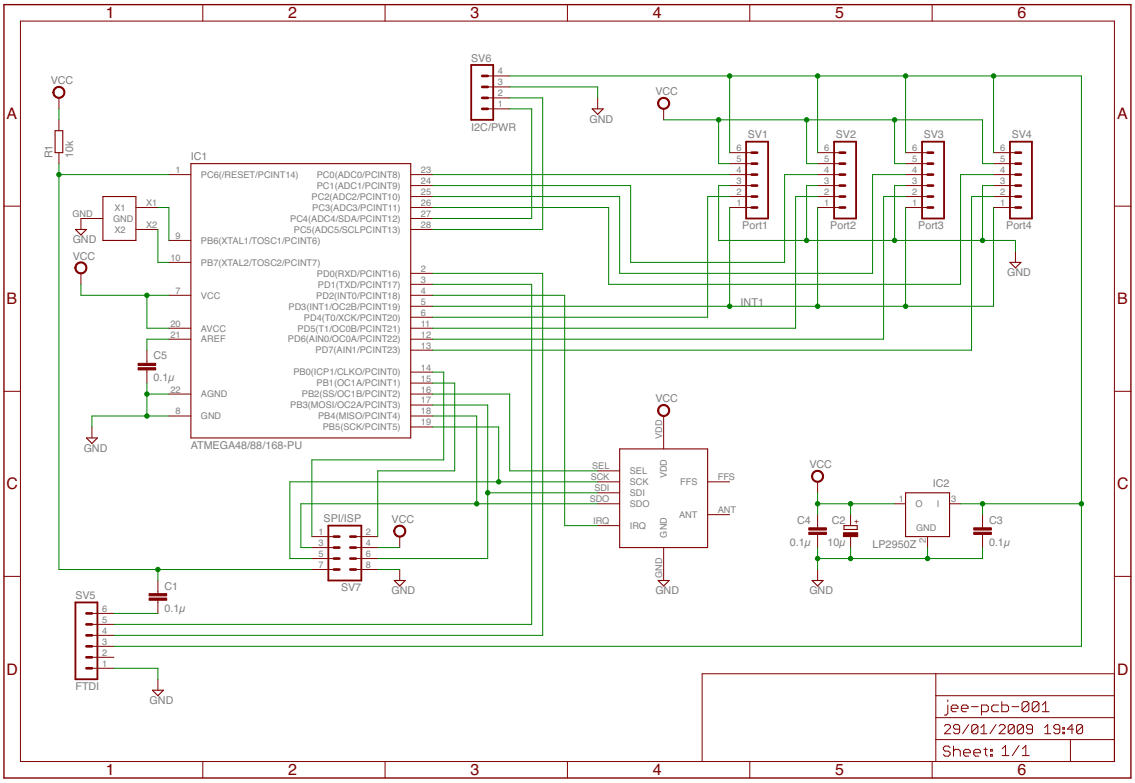
TWI/Power connector

Pin	Name	Description
1	SDA	TWI data
2	SCL	TWI clock
3	GND	ground
4	PWR	external power (+)

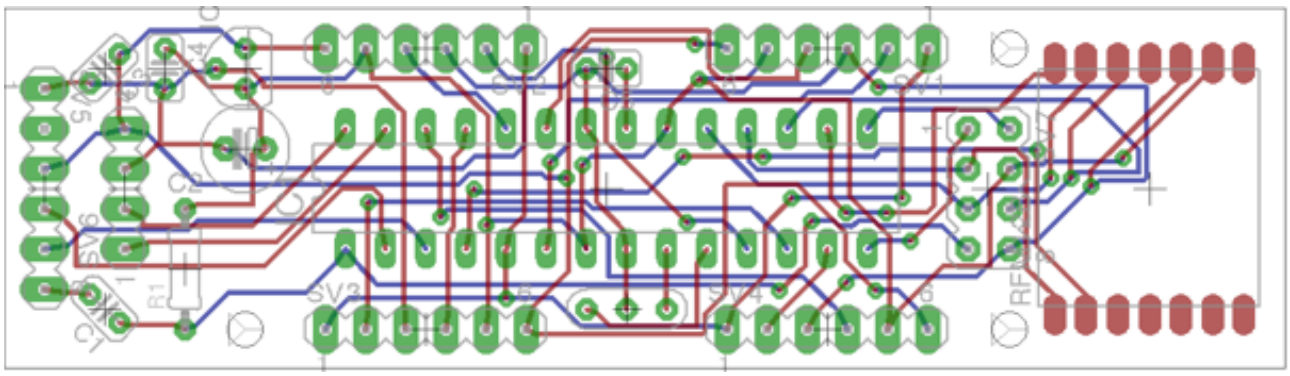
Parts list - JeeNode V1

Part	Value	Details	Digi-Key#
R1	100 kΩ	reset pull-up	
C1, C2, C3, C4	0.1 µF	decoupling	
C5	10 µF	electrolytic, polarity matters!	
IC1	ATmega168-20	could also use ATmega328	ATMEGA168-20PU
X1	16 MHz resonator	3-pin, built-in capacitors	X908-ND
IC2	MCP1702-33	3.3V LDO (alt: LP2950-3.3)	MCP1702-3302E/TO

jee-pcb-001.sch & jee-pcb-002.sch



jee-pcb-001.brd



jee-pcb-002.brd

This was a slightly different PCB layout (different holes + minor component rearrangements), but for all practical purposes it's equivalent to the jee-pcb-001 prototype board shown above.

(files created with EAGLE[4])

[4] http://www.cadsoft.de/info.htm

Document revisions

25 Feb 2009 - Initial publication.

12 Mar 2009 - Page layout & style changes. Document date *not* adjusted.