Robosoft Systems in association with JNCE presents

Swarm Robotics

www.robosoftsystems.co.in
What is a Robot

Wall-E

Asimo

ABB Superior Moti

ABB FlexPicker
What is Swarm Robotics
Let's Prepare for the Robotics Age

Powering the industries of tomorrow by rethinking what we are teaching today

Carnegie Mellon University Robotics Academy
www.education.rec.ri.cmu.edu
The ATmega16
Basic Features (1)
ATmega16 Pinout and Packages (DIP and TQFP)

www.robosoftsystems.co.in
Pin and Port Overview:

**GND:** Ground (0V)

**VCC:** Digital Supply Voltage (2.7 – 5.5V)

**AVCC:** Analog Supply Voltage

connect to low-pass filtered VCC

**AREF:** Analog Reference Voltage, usually AVCC

**/Reset:** Low level on this pin will generate a reset

**Port A, Port B, Port C, Port D:**

General Purpose 8 Bit bidirectional I/O - Ports,
optional internal pull-up resistors when configured,
output source capability: 20mA

Special Functions of the Ports available as configured using the SFRs:

**Port A:** A/D converters

**Port B:** Analog Comparator, SPI, Timer1, Timer0

**Port C:** JTAG, TWI

**Port D:** USART/UART, External Interrupts, Timer2

www.robosoftsystems.co.in
I/O Ports
I/O Ports

- General Purpose IO: Data Direction Input or Output
- Internal Pullup can be used for Input Pins
- Output driver can source 20mA current
I/O Ports

● 3 I/O-Registers for each port:

  Data Register (r/w):
  PORTA, PORTB, PORTC, PORTD

  Data Direction Register (r/w):
  DDRA, DDRB, DDRC, DDRD

  Port Input Pin Register (r):
  PINA, PINB, PINC, PIND

The Bits of these registers set the configuration for one Port Pin.
### I/O Ports – Configuration and usage

<table>
<thead>
<tr>
<th>DDxn</th>
<th>PORTxn</th>
<th>PUD (in SFIOR)</th>
<th>I/O</th>
<th>Pull-up</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>Input</td>
<td>No</td>
<td>Tri-state (Hi-Z)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Input</td>
<td>Yes</td>
<td>Pxn will source current if external pulled low.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Input</td>
<td>No</td>
<td>Tri-state (Hi-Z)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>Output</td>
<td>No</td>
<td>Output Low (Sink)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>Output</td>
<td>No</td>
<td>Output High (Source)</td>
</tr>
</tbody>
</table>

**C-Example 1** - Configure Pin B3 as output, set output level to VCC:

```c
DDRB |= (1<<3); PORTB |= (1<<3);
```

**C-Example 2** - Configure Pin D2 as input with pullup, read pin value:

```c
DDRD &= ~(1<<2); PORTD |= (1<<2); uint8_t x = PIND & (1<<2);
```
Know your Board

www.robosoftsystems.co.in
Hardware Pin outs for LED

LED Porting

LED1

1 KΩ

PB0

1 KΩ

LED2

PB1

GND = 0V
Hardware Pin outs for KEYS

Push Button Porting

Vcc = 5V

- PD2
  - S1
  - 4.7 KΩ

- PD3
  - S2
  - 4.7 KΩ
Motor Driver Pin Out

H bridge Motor Driver IC 4

Vcc = 5V
Vs = Motor Supply 12V

PD4
PB4
PB5
PD5
PB6
PB7

ENA
Input1
Input2
ENB
Input3
Input4

H Bridge1

M
Right Motor

H Bridge2

M
Left Motor

L293D

www.robosoftsystems.co.in
Motor Driver Pin out

H bridge Motor Driver IC 5

Vcc = 5V
Vs= Motor Supply 12V

PB3
PC0
PC1
PD7
PC6
PC7

ENA
Input1
Input2
ENB
Input3
Input4

H Bridge1
H Bridge2

L293D

Right Motor
Left Motor

www.robosoftsystems.co.in
The figure shown is of the IR sensor which we have used to detect the line and follow it.

The fig. b shows the function of the sensor schematically. The transmitter is a PHOTODIODE and at the receiver end it is PHOTO TRANSISTOR.
CUBE Detector IR Module

- Works on the same principle as the Grid Solving Sensor Module
- Consist of an IR pair which
- For a full white CUBE both Sensor modules will give a logic high on output
- For a half white half black CUBE only one sensor will give a logic high on output
Software Tools Required

• WinAVR – Open Source GCC Compiler
• AVR Studio – Free Atmel Debugger / Simulator
• Ponyprog – Open source Programmer
• Avrdude – Open source programmer (Reqd for USB Programmer)
Starting Avr Studio
Welcome to AVR Studio 4

Create new project

Project type:  
- Atmel AVR Assembler
- AVR GCC

Project name: test_project_1

Create initial file

Initial file: test_project_1.c

Location: C:\AVR Studio Project Test

Ver 4.18.684

<< Back  Next >>  Finish  Cancel  Help
AVR Memory Usage
-------------
Device: attiny45

Program: 134 bytes (3.3% Full)
(.text + .data + .bootloader)

Data: 0 bytes (0.0% Full)
(.data + .bss + .noinit)

Build succeeded with 0 Warnings...
Downloading through AVRDUDE

avrdude -c usbttiny -p m16  
For USB Programmer

avrdude -c siprog –P com1 -p m16  
For Serial Programmer
Response if programmer is not found
Response after connecting the programmer to the board

```
C:\Documents and Settings>avrdude -c usbtiny -p t2313
avrdude: AVR device initialized and ready to accept instructions
Reading ! ################################################################### ! 100% 0.02s
avrdude: Device signature = 0xe910a
avrdude: safemode: Fuses OK
avrdude done. Thank you.
C:\Documents and Settings>_
```
AVR Dude command to program using USB Programmer

`avrdude -c usbtiny -p m16 -U flash:w:test_leds.hex`

AVR Dude command to program using Serial Programmer

`avrdude -c siprog -P com1 -p m16 -U flash:w:test_leds.hex`
Programming through PonyProg
Led Blink Program (Software Delay)

// LED Blink Program using Software Delay

// #define F_CPU 8000000UL /* 8MHz CPU clock */
#include <util/delay.h>
#include <avr/io.h>

int main(void)
{
    DDRB = 0xff; // Make Port B as Output

    for(;;) // endless loop
    {
        PORTB = 0x55; // led1 ON & led2 OFF
        _delay_ms(500); // Wait for delay of .5sec.

        PORTB = 0xaa; // led1 OFF & led2 ON
        _delay_ms(500); // Wait for delay of .5sec.

    }

    // Never reached.
    return(0);
}
End of Day 1
RF Module

Features:

- Supply voltage of 5V to 12V.
- Operates in ISM [industrial, scientific and medical and Short Range Devices] Frequency at 2.4-2.4835GHz.
- Supports 256 channels i.e. From 0-255.
- Supports 255 devices per channel.
- Supports variable BAUD rate.
- RS232 compatible.
Jumper Setting

- **CONFIG MODE**
  - Closed: Configuration mode
  - Open: Run mode

- **PACKET MODE**
  - Closed: Variable Packet Length (with device address selection)*
  - Open: Single Byte Transfer (Broadcast) (80msec delay between 2 char)

*Note: To switch between modes, you have to power on reset module

www.robosoftsystems.co.in
## Configuration Mode:-

<table>
<thead>
<tr>
<th>Sr.</th>
<th>Entity</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Channel Number</td>
<td>0</td>
<td>255</td>
</tr>
<tr>
<td>2</td>
<td>Device Address</td>
<td>0</td>
<td>255</td>
</tr>
<tr>
<td>3</td>
<td>UART baud rate</td>
<td>300 bps</td>
<td>9600 bps</td>
</tr>
</tbody>
</table>
## RUN Mode:

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Data Transmit Receive Mode</th>
<th>Min Length</th>
<th>Max Length</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Single Character Mode</td>
<td>1</td>
<td>1</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>Packet Mode</td>
<td>2</td>
<td>40</td>
<td>variable</td>
</tr>
</tbody>
</table>
## Packet structure

<table>
<thead>
<tr>
<th>#</th>
<th>Packet length</th>
<th>Device ID</th>
<th>Data field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character</td>
<td>Purpose</td>
<td>Packet length</td>
<td>= Device ID + Data field (Range: 0x02 – 0x40)</td>
</tr>
<tr>
<td>#</td>
<td>Start of packet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device ID</td>
<td>Identifies the device (Range: 0x00 – 0xFF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data field</td>
<td>Data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Channel and Device ID config:-
To choose and select among 256 channels and 256 devices per channel
PORT config:-
To choose and select available com ports
If you have already chosen the required port then the TAB turns GREEN indicating that the port is available but not open.
If you have already OPENED the selected port then the TAB turns RED indicating that the port is not available and after its use you can close the port.
If you have not already chosen the required port then the TAB turns RED indicating that the port is not selected and the TAB is DISABLED.
If you have any problem in using this software then use the HELP tab for relevant topic.
AVR Fuse Calculation
http://www.engbedded.com/fusecalc
2.4 GHz RF Module for Wireless Communication (Designed for Nexus Competitions)

Be the first to review this product

Availability: In stock.

₹ 600.00

Qty:  Add to Cart  OR  Add to Wishlist
Add to Compare

Quick Overview

This RF module is a plug and play replacement for the wired Serial Port (UART) especially designed for the Nexus Competitions. This RF module allow engineers of all skill levels to quickly and cost-effectively add wireless capabilities to virtually any product.

Downloads

RF Module Manual v2.pdf (1026 Kb)
RF Module Configuration.zip (5000 Kb)
Lab Products: Self Balancing Bot
Call for more details....

Sensor Modules

- Accelerometer Breakout Board - ADXL322 +/-2g
  - Price: Rp 3,000.00
  - Out of stock
  - Add to Wishlist
  - Add to Compare

- Digital Compass Module - HMC6352
  - Price: Rp 2,000.00
  - Out of stock
  - Add to Wishlist
  - Add to Compare

- GPS MiniMod (TIL)
  - Price: Rp 1,500.00
  - Out of stock

Visit our Robot Kits section for more...