12Blocks

12Blocks v1
October 2009

12Blocks Manual

by Hanno Sander

Table of Contents

Welcome

1.1 Welcome

12Blocks: ***"What can you build with a dozen blocks?"***

Welcome to 12Blocks, the best tool for projects with the Parallax Propeller.

What is 12Blocks

12Blocks is a tool that let's you create programs by stacking blocks together. A library of over 100 powerful blocks let's you quickly build games, robots, and more. Just assemble blocks onto the worksheet and then debug your program with built-in graphical tools. 12Blocks is ideal for classrooms, students, hobbyists and professionals who want to quickly and easily build projects with the Parallax PropellerTM.

How can 12Blocks help me

- write powerful programs by assembling blocks onto a worksheet
- graphically debug programs with the built in visualizers
- use wizards to customize graphical sprites, vectors and more for your program
- change program parameters on the fly- without recompilation/reloading
- interface with common sensors, peripherals, actuators and devices
- easily integrate the Propeller with other PC applications
-

Sample Blocks:

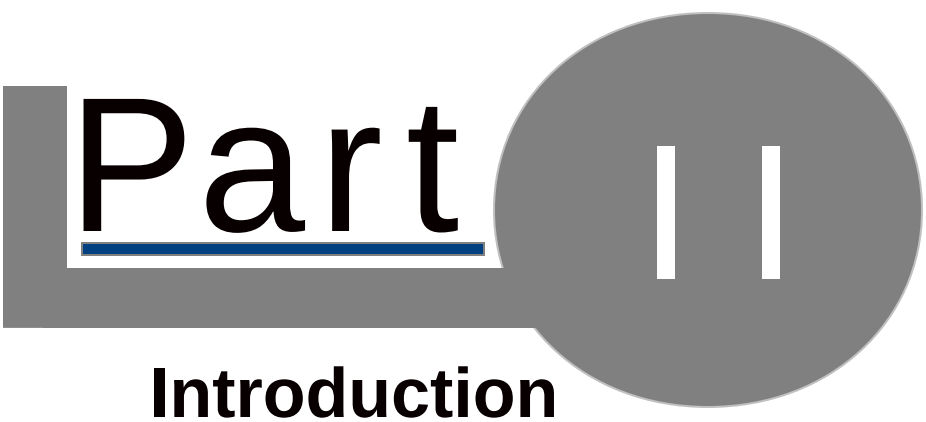
- speak "hello" (speech synthesis)
- draw sprite "dog": (TV/VGA graphics)
- record sound for "1" seconds (record WAV files)
- ramp servo "1" to "100"% (servo control with ramping)
- brightness on "2" (interface with sensors)
- get mouseX (interface with PS2 mouse/keyboard)
- measure frequency on pin "4" (IO input)
-

Features:

- Graphically arrange blocks to create a program- no typing or remembering cryptic commands
- Self contained blocks manage memory, objects, code, cogs for you
- Easily edit block parameters- wizards exist for vectors, sprites, speech, wav files
- Change value parameters while your program is running
- Use multiple "start" blocks to write a multiprocessing program
- Comment your program with blocks than include links to files or websites
- Intuitive flow control/event notification with Broadcast and Receive blocks
- Use "Add Block" to add additional blocks to the library, custom libraries can be shared
- Use Hardware definition files to define pin usage and restrict library to what is supported
- Share programs online with the 12Blocks community
- Unlimited undo/redo buffer
- Program looks and feels like code and saved as standard Parallax spin file- eases transition
- Arbitrary parameters- combine blocks and text
- View and change variable values in a running program using an oscilloscope interface
- View pin activity using a graphical map and logic analyzer graph
- Built in terminal
- Integrate with Python, Matlab, Excel, VB.NET, C# or other DDE/.NET aware applications
- Comprehensive Help Manual and context sensitive help, Easy Install/Uninstall, tutorials
-

Links:

Home Page:	http://12blocks.com
Blog:	http://blog.mydancebot.com
Forums:	http://forums.mydancebot.com



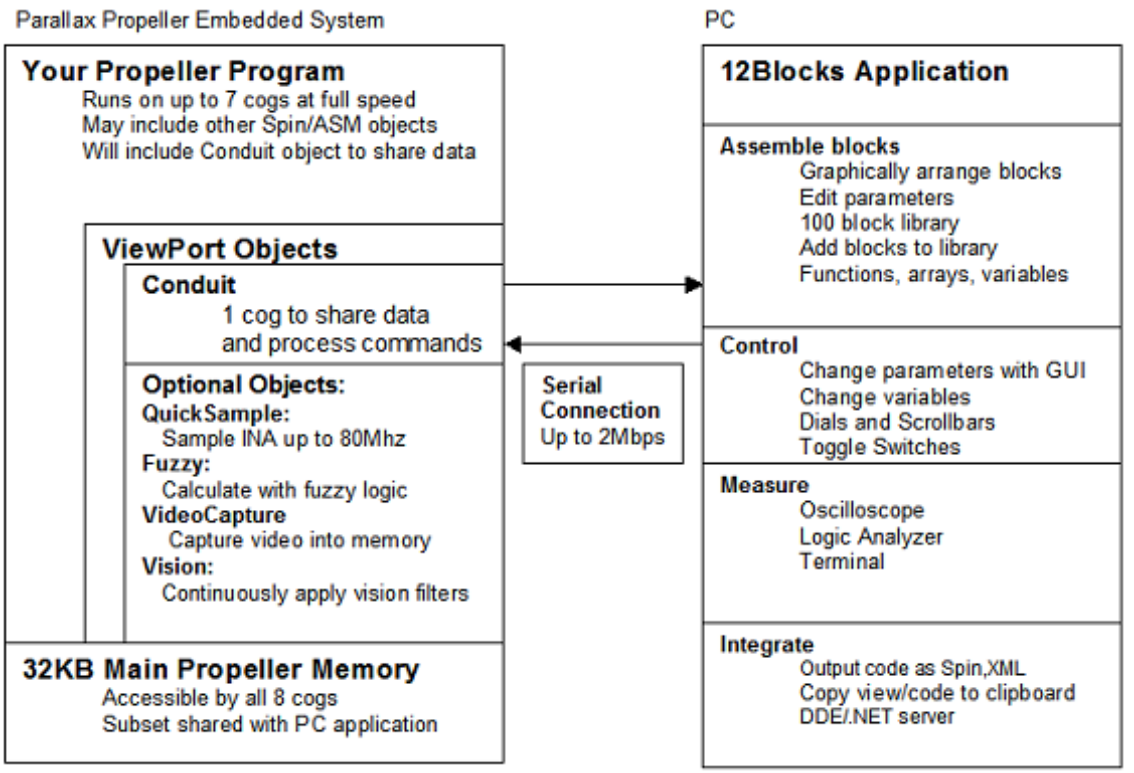
Introduction

2.1 Overview

Background

The Parallax Propeller is a powerful microprocessor used in embedded systems, robotics and advanced electronic projects. 12Blocks is a graphical tool that let's users build programs for the Propeller by assembling blocks. 12Blocks compile the users' program to code that runs on the Propeller. 12Blocks also uses a serial connection to the Propeller to monitor and change data in the user's program as it runs.

12Blocks Architecture



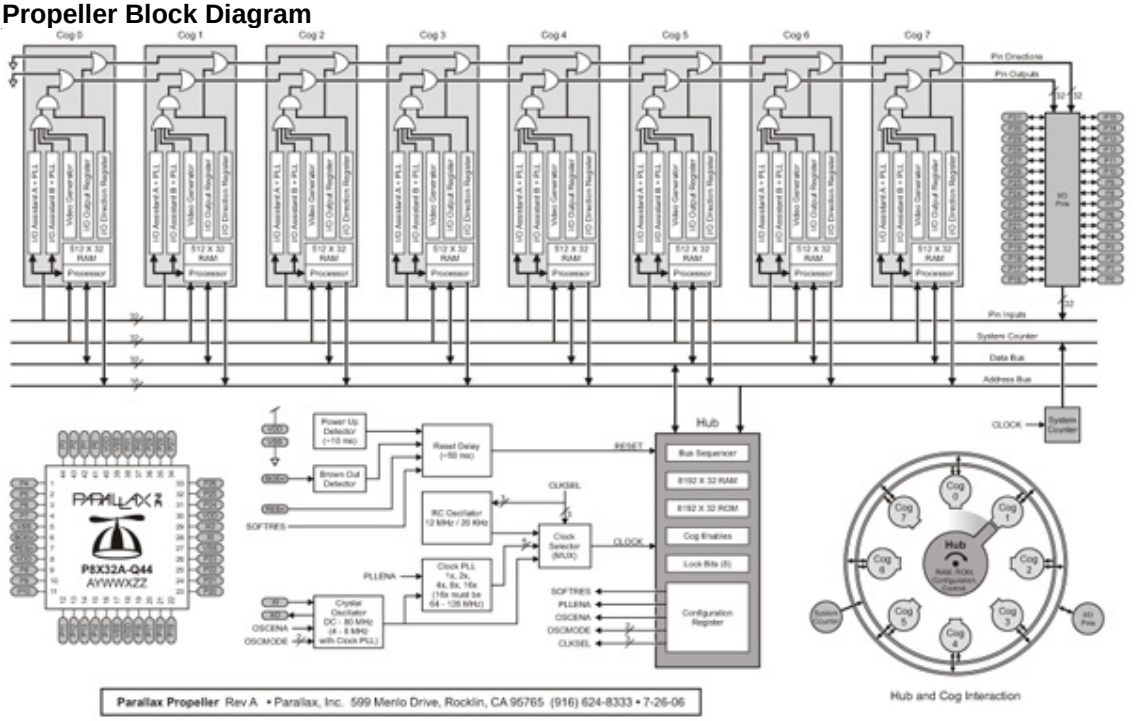
2.2 Parallax Propeller

Parallax Propeller: 32 Bit, 80MHz, 8 cog Microprocessor
The Propeller chip makes it easy to rapidly develop embedded applications. Its 8 processors (cogs) can operate simultaneously, either independently or cooperatively, sharing common resources through a central hub. The developer has full control over how and when each cog is employed; there is no compiler-driven or operating system-driven splitting of tasks among multiple cogs. A shared system clock keeps each cog on the same time reference, allowing for true deterministic timing and synchronization. Two programming languages are available: the easy-to-learn high-level Spin, and Propeller Assembly which can execute at up to 160 MIPS (20 MIPS per cog).

Propeller Specification:
Processors: 8 32 bit processors running at up to 80MHz
Shared Global Resources: 32KB RAM, 32KB ROM, 32 IO Pins
Dedicated Resources Per Processor: 2KB RAM, 2 General Counters, Video Output
Power: 3.3volt DC, each pin can sink up to 40mA

Propeller Development Environment:
The Propeller Tool is a free Editor/Compiler/Loader for the Spin and Assembly languages. Compiled programs are loaded to the Propeller over a serial connection.
A large community of users frequent the Parallax Forums: http://forums.parallax.com
The repository of source code object is known as the Object Exchange: http://obex.parallax.com
Information about the Parallax Propeller can be found here: http://www.parallax.com/propeller

Propeller Block Diagram



Courtesy of Parallax Inc.

2.3 Uses

12Blocks is ideal for anyone starting out with the Propeller- it makes it easy to take full advantage of all the Propeller's capabilities. Transitioning to programming in spin with the Propeller Tool is easy as 12Blocks opens/saves regular spin files.

A perfect hardware complement is the Propeller Demoboard to let you:
- drive a robot with servos
- draw graphics for vga/tv
- react to keyboard/mouse
- communicate with a terminal
- output music tones, wav files, record and play back sounds, and synthesize speech
- interface with sensors like the PING and a compass
- perform input/output on the pins: measure frequency,pulse, count edges...

Use Case	Detail
PC-based Controller	Connect Propeller to a device and use 12Blocks to monitor and change the variable controlling the device- for example, the position of a servo.
Tuning and Calibrating Parameters	Use a variable in 12Blocks to change the parameters of a PID Control
Interface with other Hardware	Use the LSA view to monitor timing signals when working with I2C devices like a compass or eeprom
Take Measurements	View and measure the signals from an ADC with "View Values"
SCADA Prototype Tool	Control a remote system and log data for sharing with other programs.
Teaching Tool	Introduce operation of basic instrumentation such as DSO, LSA etc.
Robot Design Platform	View sensor values and control actuators with 12Blocks

Projects Ideas*	Detail
Internet Weather station	Add sensors to the Propeller: temperature, wind direction and speed, camera.. Plot the measurements in "View Values"
Balancing Robot	1. Build robot with tilt sensor and wheel encoder 2. Tune the control algorithm
Digital Oscilloscope/Spectrum Analyzer	1. Add an ADC to the Propeller to measure analog values 2. View signal in "View Values"
Function Generator	1. Add a DAC to the Propeller to generate analog values 2. Assign controls to control the waveform: pwm, sine, square...
Measure sensors	Gyro, accelerometer, Ping, IR , wheel encoder, compass, GPS, temperature …….
Control actuators	Solenoid, Hobby Servo, motors, stepper motor, H-bridge
Process Control	Measure temperature and pH while fermenting beer

*For additional ideas visit our site: http://12blocks.com/howto to see how people use 12Blocks.

2.4 About myDanceBot

The people behind myDanceBot.com believe in building sophisticated, yet affordable products by pushing off the shelf components to their limits. We aim to understand the components we use through detailed measurements and analysis- then we engineer the optimal solution.

Thus far, we have developed:

- ViewPort - the premier debugging environment for the Parallax Propeller
- 12Blocks- the block based programming environment for the Propeller
- DanceBot, a balancing robot that can dance
- IODreamKit, which replaces a lab's worth of instruments with a single board
- Software for the PropScope, Parallax's next generation USB Oscilloscope
- Spin Code Debugger, a fully integrated graphical debugger for SPIN code
- a Conduit object that moves data to/from the PC at 2Mbps through optimized bitbanging
- a QuickSample object that samples the IO port every cycle with advanced self-modifying
- code running interleaved on 4 cogs
  a high performance Fuzzy Logic engine with integrated Control Panel
- OpenCV Integration, making it easy to integrate OpenCV vision technology with Propeller
- projects

Links:

Home Page:	http://mydancebot.com
Blog:	http://blog.mydancebot.com
Forums:	http://forums.mydancebot.com
Parallax Propeller:	http://parallax.com/propeller

Getting Started

3.1 Installation

Requirements:
PC host system:
>500MHz CPU with 5MB HDD Space, 500MB RAM, Mouse
- Windows 95,98,2k,XP,Vista
- USB 2.0 connection (preferred) or serial to Propeller
- Parallax Propeller Tool software installed with USB-serial driver installed
-

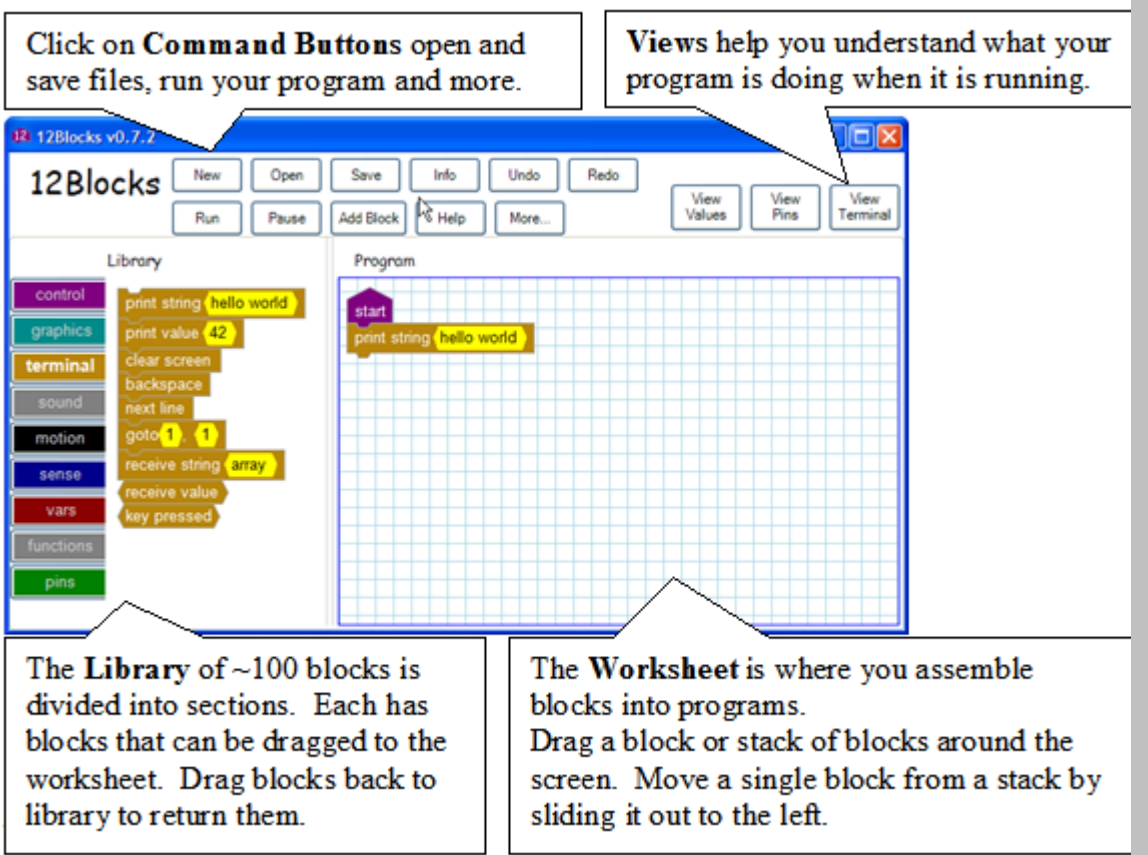
Parallax Propeller target
Parallax propeller chip with power supply
- 5MHz crystal
- EEPROM, LEDs, video input are optional
- The Parallax ProtoBoard is a low-cost, high-quality solution for Propeller projects
-

Installation:
Download 12Blocks and follow the installation wizard to install it. If you're upgrading, the installer will replace old 12Blocks files with the updated versions. Before starting 12Blocks, make sure a Parallax Propeller is connected to your computer- ideally with a USB connection. USB allows for faster connection up to 2Mbps.

Uninstall:
To uninstall, select the uninstall item from the Windows Start menu.
Video Tutorials:
The video tutorials show ViewPort in action- view them here: http://12blocks.com/videos
Problems:
If you encounter problems, check the Problem Solving guide or contact us:
http://mydancebot.com

3.2 Interface Overview

12Block's interface consists of Command Buttons, View, a Library and a Worksheet:



Your programs must start with one of the *start* blocks- they're drawn with a triangular top. So start your program by dragging the *start* block to the worksheet. (Move your mouse to the *star* block in the library, hold down the left mouse button, move the mouse back to the worksheet and release the mouse button).

3.3 Blocks

Drag a block from the library to the worksheet to use it.

- Programs must start with “Start” blocks- indicated by their triangular top.
- Programs can include multiple start blocks
- Many blocks take parameters- indicated by yellow boxes. Click on the yellow area to enter text.
- Blocks with triangular sides can only be used as “results”- they must be dragged on top of a yellow parameter area.
- Most other blocks can be used as “result” blocks- drag them onto a parameter area.
- Parameter areas can include both text and blocks
- Right click on a block to copy or delete it
- Right drag a block to create a copy of it
- You can create your own blocks with the "Add Block" button
- Remove a block by dragging it back to the library
-

3.4 Parts of a Program

- Start Blocks (indicated by triangular tops) are used to start a stack of blocks.
- Blocks have some effect- they do something. Most can be customized by their yellow parameter areas. Some have multiple attachment tabs.
- Loop blocks repeatedly execute their stack
- Conditional blocks execute their stack if a condition is met
- Expressions can be typed into a parameter area. An expression can include text and other blocks. Expressions can include variables and arrays. Expressions are evaluated in a specific order. Expressions can be Boolean or produce a value.
- Variables can be global to the entire program. Variables can be part of the argument list of a function. Variables can be local to a function. Edit global variables in the "vars" section.
- Arrays are global and can be used for strings- edit them in the "vars" section.
- Functions are defined using a "Function Start" block. You need to specify the function's name, arguments and local variables in the start block. You can use a return block to return a value. The function section of the library will contain a block to call your function once you've named it. Changing the name or arguments of your function's start block wil change all blocks that call that function. You can call a function from itself- this is recursion.
- Messages can be sent and waited on to deal with synchronous and asynchronous events. Start a stack with a "Wait for Message" block to handle that event. Use the "Send Message" block to start an asynchronous event- the code in the handler will run in parallel. Use the "Send Message and Wait" block to start a synchronous event- the code in the handler will run and finish before your code resumes.

3.5 Peripherals

- Terminal: Use blocks from the “terminal” library section to write text or values to the built-in terminal. You can also position the cursor, clear the screen, or set the position. You can input a string of text or wait for specific characters to be typed
- Keyboard: Use the “wait for key” block to handle a specific keypress on a keyboard connected to the Propeller. Or use the “key” block from the “sensor” section to detect if a particular key is pressed.
- Mouse: Use the “wait for click” block to handle a click from a mouse connected to the Propeller. Or use the “mouse-pos” blocks from the “sensor” section to read the x,y,z position of the mouse.
- Graphics: Use the “text” blocks from the “graphics” section to print characters to a monitor. Or, draw shapes, sprites, vectors and text using the “graphics” blocks from the same section. Under “More..” you can choose if your Propeller is connected to a TV or a VGA monitor.
- Sounds: Use the “audio” blocks to create sound from tones, synthesized speech, or wav files.
- Motion: Use the “motion” blocks to control servos and motors. The “go” block lets you control the speed and turn rate with one block.
- Sense: Use the “sense” blocks to interface with sensors like the qti line sensor or an IR distance sensor.
- Pins: Use the “pins” blocks to perform all sorts of low-level pin input and output.

3.6 Debug with Views

12Blocks includes Views to help you debug your Program

- "View Values" shows you the value of global variables. You can plot variable values versus time at selectable time scales. Drag the graph to move the trace. Click to the left of the graph to set a trigger.

- "View Pins" shows the IO state of the Propeller's 32 IO pins. Red indicates the pin is currently "high", or 3.3V, blue indicates it's "low", or 0V. The "Logic State Analyzer" indicates activity for pins used by your program versus time at selectable time scales. Drag the graph to move the traces. Click to the left of the graph to set a trigger

- "View Terminal" shows the terminal which shows output and allows input to the Propeller.

3.7 Tutorial

Pre-requisites:

- 12Blocks should be installed
- Propeller turned on and connected to this PC
-

Tutorial

Start 12Blocks and at the Welcome Screen, select "**Learn from the Tutorial".** You can also download the tutorial here: http://12blocks.com/tutorial.pdf

The tutorial will walk you through several examples that increase in complexity- from a simple "Hello World" program to calculating Fibonacci sequences with arrays and recursive functions.

3.8 Sample Programs

Pre-requisites:

- 12Blocks should be installed
- Propeller turned on and connected to this PC
-

Sample Programs

Start 12Blocks and at the Welcome Screen, select "Open and Existing Program**".** Select one of the sample programs and press "Open". You're "worksheet" will now show the program you've loaded. Press "Info" to get more information about it, or "Run" to run it on the Propeller.

Part IV

Reference (Windows Application)

4.1 Command Buttons

New-click to start a new program
- Open-click to open a program
- Save-click to save the program
- Save As-click to change the program file's name. Can save as:
- .12b: the default for 12Blocks
  - .spin: compatible with the Propeller Tool
  - .png: a graphical image of the code
  - .xml: a computer readable format of the program
  - .zip: an archive for sharing the program and it's objects with others
- Info-click to view information about the current program- including schematics, description
- and images. Includes button to share the program with others online

Undo- undoes the last action
- Redo- redoes the last action
- Run- compiles and loads the current program
- Pause- disconnects the serial connection with the hardware thereby stopping new "view" data
- from being graphed

Add Block- click to start a wizard to add a new block to 12Blocks
- Help- displays this help file
- "More" Menu
- Load to EEPROM- click to load the current program into the Propeller's EEPROM
  - View Code- click to view the spin code for this program in the Propeller Tool
  - Print- click to print the code to a printer
  - Copy Code to Clipboard- click to copy an image of the code to the clipboard
  - Copy View to Clipboard- click to copy an image of the last used view to the clipboard
  - VGA/TV Graphics- click to select which type of graphics the Propeller will output
  - Ports- click to select which ports will be searched for the Propeller
  - Hardware- click to select from different hardware targets. A target file specifies the clock
  - rate and pin layout.

    Report Bug- click to report a bug to the author to get it fixed
  -

4.2 Folder Structure

The 12Blocks installer checks for a valid .net Framework installation, presents the license agreement and installs files as follows:

Program Files/12Blocks
- **12Blocks.exe**
  - core application
- **unins000.dat/unins00.exe**
  - uninstall files
- **vplib.dll**
  - core library
- **help.chm**
  - Help document
- **excel client.xls**
  - Sample excel worksheet which interfaces with 12Blocks

Program Files/12Blocks/Plugins
- **.dll:** Additional plugins

Program Files/12Blocks/View
- **.xml files:** layout data for each view
- **hierarch of image files:** files used by views

My Documents/12Blocks
- **.12b files:** store your program files here
- **.spin files:** spin object files
- **.say, .vector, .sprite, .wav files:** object files

My Documents/12Blocks/Tutorials
- **.12b files:** sample 12b files
- **.spin files:** spin object files
- **.say, .vector, .sprite, .wav files:** object files

4.3 Tutorials

The following tutorials are included with 12blocks:

Graphics
This program continually draws vector and sprite drawings to a TV or monitor. It also draws some text and a triangle. The loop changes the size of the vector drawing.

Lights
This program uses 3 start blocks to blink 3 LED's. The first uses a toggle block combined with a wait. The next uses a Frequency output block. The last uses the pwm block to slowly brighten an LED over several seconds.

Servos
This program moves a servo to it's end and then slowly ramps it to the other end. It also drives a robot forward, turns, then drives while turning.

Sounds
This program plays tones, uses a software speech synthesizer to speak a word, and then plays a WAV file.

Sensors
This program displays readings taken from some sensors

4.4 Connection Properties

Port
12Blocks uses the Parallax Propellent module to compile spin files and manage the connection with the Propeller. You can configure the order ports will be scanned and which ports to skip using the Propellent configuration screen available from the "Ports" item of the "More" menu.

Baud Rate
12Block's default baud rate is 1MBaud- equivalent to 1 million bits/second or roughly 100,000 bytes/second. This allows 12Blocks to quickly transfer data back and forth to the Propeller to support high sampling rates as well as streaming video.

4.5 Terminal

12Blocks includes a terminal view that shows the terminal output of your program when using the terminal blocks. Entering text into the text area will send the keystrokes to the Propeller where they can be read in with the terminal input blocks.

4.6 Pause

The Pause button let you start or stop the connection with the Propeller. You can still view, manipulate, and save data, but no new data will be captured while the connection is stopped.

Every time you start a connection with the Propeller, 12Blocks will query the program's configuration. If the program's configuration has changed (because you loaded a new program), then 12Blocks will use that new configuration to it receives from the Propeller to configure itself. Configuration can also be saved and loaded as files or included in a program.

Only one Program on your PC can use the COM/USB port that connects the PC to the Propeller chip.

Problem Solving

5.1 Problem Solving

- **Setup.exe doesn't run:**
  Make sure you have sufficient priviledges to install a new application. Also insure that your Antivirus software is not blocking new installs.

- **Setup tells me to install the Microsoft .Net Framework:**
  This is normal. 12Blocks was built on .Net Framework. It should detect if you need to install this, and will then help you download and install it from Microsoft.

If you have other problems try our forums: **http://forums.mydancebot.com**

Part VI

How To Buy

6.1 How To Buy

Free Evaluation
Download a 30 day free trial version of 12Blocks from our website: http://12Blocks.com

Purchase License Online
Visit our website to securely purchase a license for the components you wish to use using Google Checkout or PayPal:
http://12Blocks/register.php

Upgrade your License Online
Visit our website with your existing license to upgrade your license to use additional components.
http://12Blocks/upgrade.php

Distribution:
Distribution of 12Blocks Trial are allowed. Distribution of licenses is NOT allowed. 12Blocks may not be installed on a network.

Copyrights:

KEYWORD INDEX