ARDUPILOT SITL SETUP FOR WINDOWS

Guide to setting up SITL simulations in windows

NOVEMBER 5, 2017

SCHULICH UAV
University of Calgary

Contents

Introduction

The purpose of this guide is meant to serve those who wish to do a SITL (Software in the Loop) simulation. Effectively a simulation version of Ardupilot firmware will be compiled and executed. The advantage of doing the simulation in windows is that it allows the simulation to connect to Mission Planner which is more feature rich than alternative ground stations available in Linux. This test setup was based on the guide available at http://ardupilot.org/dev/docs/sitl-native-on-windows.html , however a few things needed to be changed as the documentation is out of date. SITL was successfully built on Windows 10.

Instructions

This section of the document outlines the procedure for getting the SITL simulation running.

Installing MavProxy

The first thing to do is to install MavProxy, a light weight ground station/proxy necessary to connect the SITL to Mission Planner later.

1. Download the latest Windows version of MavProxy available here http://firmware.ardupilot.org/Tools/MAVProxy/
2. Install MavProxy through its install wizard (no special settings as you essentially just need to 'click through it')

Installing Mission Planner

Mission Planner is a free, comparatively feature rich ground station software available only for Windows. It is a very handy tool and not to mention our main ground station of choice. Follow the instructions below to install it

1. Download Mission Planner from the following link http://firmware.ardupilot.org/Tools/MissionPlanner/MissionPlanner-latest.msi
2. Install Mission Planner through its install wizard (no special settings as you essentially just need to 'click through it')

Installing Flight Gear

Flight Gear is a visualization/modeling tool that provides a viewer for watching the simulated vehicle fly as it is running a mission. These instructions provide a guide on how to set it up.

1. Download FlightGear 3.4.0 via the following link https://osdn.net/frs/g_redir.php?m=netix&f=%2Fflightgear%2Frelease-3.4.0%2FSetup+FlightGear+3.4.0.exe
2. Install FlightGear through the installation wizard(no special settings as you essentially just need to click through it)

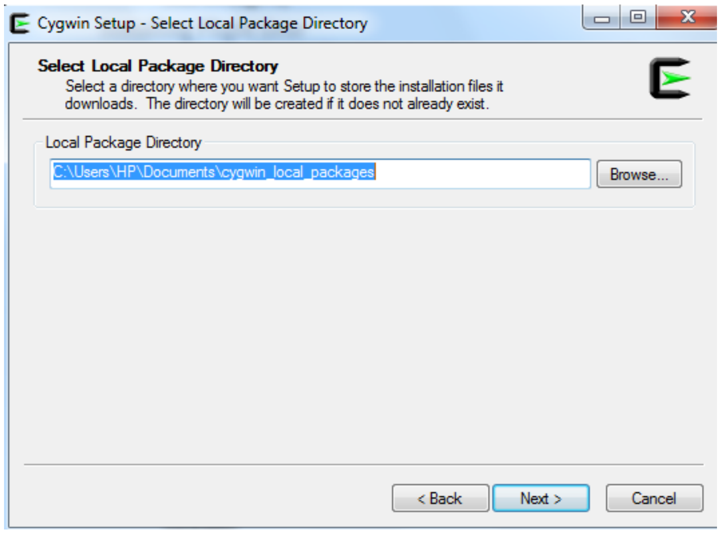
Installing Cygwin and Setting up SITL

Cygwin is a tool which provides a 'linuxish' terminal interface on Windows as well as providing access to needed linux packages on Windows. Follow the instructions to set it up. This guide was developed and tested using the 64 bit version of Cygwin.

1. Download the appropriate version of Cygwin for your computer's architecture

https://cygwin.com/install.html

2. Start the installation **but do not blindly walk through** and I would recommend making an appropriate folder when it asks for where its local packages should be installed



Cygwin Setup - Select Local Package Directory

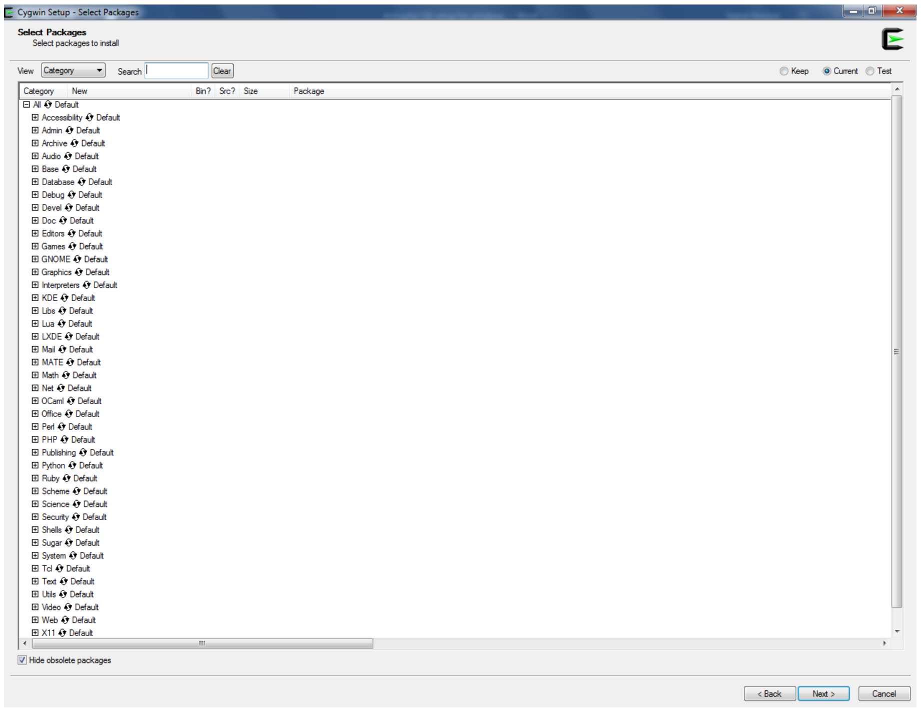
Select Local Package Directory

Select a directory where you want Setup to store the installation files it downloads. The directory will be created if it does not already exist.

Local Package Directory: C:\Users\HP\Documents\cygwin_local_packages

Figure 1: The part of the Cygwin Installation where it asks where local packages should be installed

3. Install the packages listed on the table below when the installer asks what packages need to be installed (triple check that you have the indicated packages before proceeding)



Cygwin Setup - Select Packages

Select Packages

Select packages to install

View: Category   Search: ( ) Keep (●) Current ( ) Test

```
Category   New                    Bin?  Src?  Size   Package
⊟ All Default
  ⊞ Accessibility Default
  ⊞ Admin Default
  ⊞ Archive Default
  ⊞ Audio Default
  ⊞ Base Default
  ⊞ Database Default
  ⊞ Debug Default
  ⊞ Devel Default
  ⊞ Doc Default
  ⊞ Editors Default
  ⊞ Games Default
  ⊞ GNOME Default
  ⊞ Graphics Default
  ⊞ Interpreters Default
  ⊞ KDE Default
  ⊞ Libs Default
  ⊞ Lua Default
  ⊞ LXDE Default
  ⊞ Mail Default
  ⊞ MATE Default
  ⊞ Math Default
  ⊞ Net Default
  ⊞ OCaml Default
  ⊞ Office Default
  ⊞ Perl Default
  ⊞ PHP Default
  ⊞ Publishing Default
  ⊞ Python Default
  ⊞ Ruby Default
  ⊞ Scheme Default
  ⊞ Science Default
  ⊞ Security Default
  ⊞ Shells Default
  ⊞ Sugar Default
  ⊞ System Default
  ⊞ Tcl Default
  ⊞ Text Default
  ⊞ Utils Default
  ⊞ Video Default
  ⊞ Web Default
  ⊞ X11 Default
```

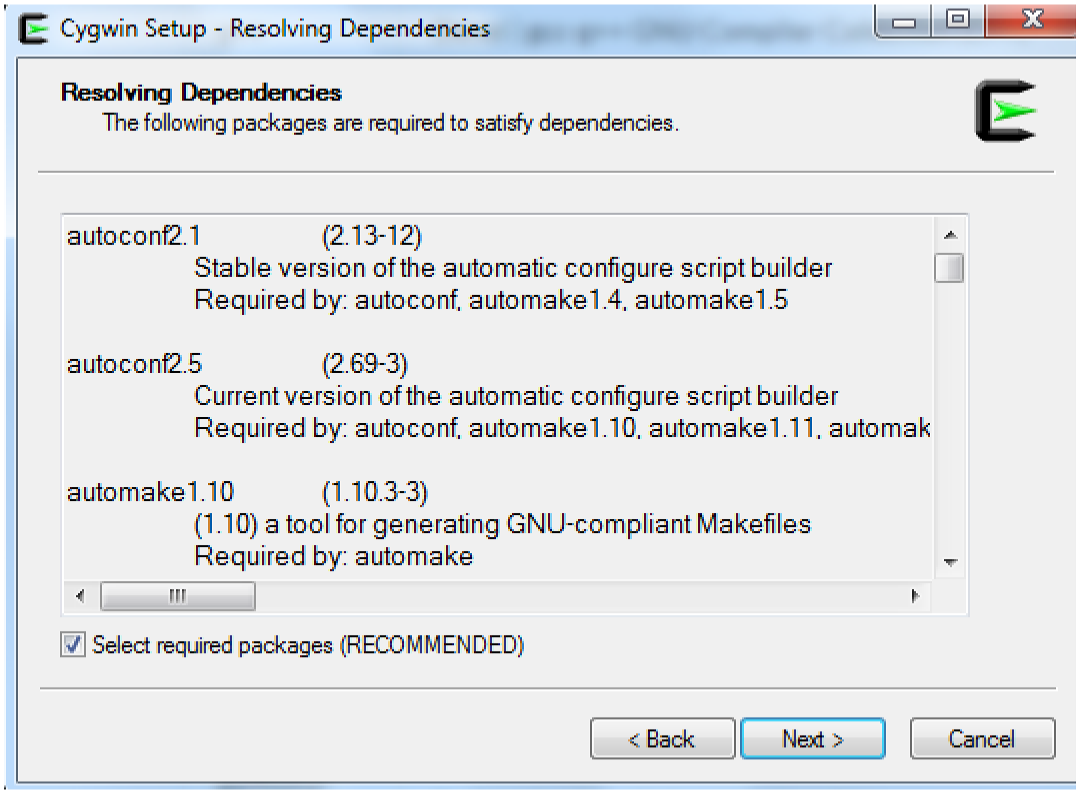
[✓] Hide obsolete packages

Figure 2:The part of the install process where one must install desired packages for Cygwin

Table 1: Packages that must be installed for Cygwin

Package Name	Category	Description
autoconf	Devel	autoconf: Wrapper scripts for autoconf commands
automake	Devel	automake: Wrapper for multiple versions of automake
ccache	Devel	ccache: A C compiler cache for improving recompilation
g++	Devel	gcc-g++ GNU Compiler Collection (C++)
git	Devel	git: Distributed version control system
libtool	Devel	libtool: Generic library support script
make	Devel	make: The GNU version of the 'make' utility
gawk	Interpreters	gawk: GNU awk, a pattern scanning and processing language
libexpat	Libs	libexpat-devel: Expat XML parser library (development files)
libxml2-devel	Libs	libxml2-devel: Gnome XML library (development)
libxslt-devel	Libs	libxslt-devel: GNOME XSLT library (development)
python2-devel	Python	python2-devel: Python2 language interpreter
procps	System	procps-ng: System and process monitoring utilities
dos2unix	Utils	dos2unix: Line Break Conversion

4. Install the required dependencies when asked



Resolving Dependencies
The following packages are required to satisfy dependencies.

```
autoconf2.1        (2.13-12)
        Stable version of the automatic configure script builder
        Required by: autoconf, automake1.4, automake1.5

autoconf2.5        (2.69-3)
        Current version of the automatic configure script builder
        Required by: autoconf, automake1.10, automake1.11, automak

automake1.10       (1.10.3-3)
        (1.10) a tool for generating GNU-compliant Makefiles
        Required by: automake
```

☑ Select required packages (RECOMMENDED)

Figure 3: After selecting the packages required by the table, accept when asked to install required depedencies

5. Once Cygwin is installed open a terminal and enter the following command

   ```
   git clone https://github.com/sojhalb/sitl_windows.git
   ```

6. Enter the cloned directory and enter convert the files to Unix format and run the script (Note this step requires a significant amount of memory and it is recommended that you close as many other processes, applications, etc as possible before running the script)

   ```
   cd sitl_windows
   dos2unix setup.bash
   ./setup.bash
   ```

   If MavProxy starts then the script likely successfully executed

Operation

This section of the document shows you how to run the simulation.

1. (OPTIONAL) Initiate FlightGear by opening a file explorer and heading over to the *'C:\cygwin64\home\<USERNAME>\ardupilot\Tools\autotest'* directory and launching the appropriate Flight Gear Script.

   For example to launch the Plane view launch the *fg_plane_view* script, for a quadcopter launch the *fg_quad_view* script

2. Open Cygwin and head over to the autotest directory

   ```
   cd ~/ardupilot/Tools/autotest/
   ```

3. Execute the SITL simulation with the desired options

   ```
   ./sim_vehicle.py -v ArduPlane -j4 -L KSFO
   ```

   The above command starts the ArduPlane simulation and sets the Plane at San Fransisco International Airport

4. Open Mission Planner and connect through UDP at port 14550