

# SMARTBASIC APP DOWNLOAD PROTOCOL OVER UART

Application Note

v1.0

## INTRODUCTION

The goal of this document is to describe the protocol used to enable an embedded host processor to load a compiled *smartBASIC* script to any of Laird's *smartBASIC* enabled modules, such as the BL600, BL620, and BT900, over the UART.

## REQUIREMENTS

The following equipment and utilities are required:

- Windows PC
- *smartBASIC*-enabled module
- Xcomp\_nnnn\_hhhh\_hhhh.exe (nnnn and hhhh\_hhhh changes depending on firmware)
- A *smartBASIC* '\*.sb' application
- Space in Host-accessible flash to store the resulting '\*.uwc' compiled script

## COPYING THE COMPILER .UWC FILE INTO THE MODULE'S FILE SYSTEM

To download a *smartBASIC* app into the module, follow these steps:

1. Using the xcomp\_nnnn\_hhhh\_hhhh.exe cross-compiler that Laird provides, compile the *smartBASIC* .sb file into a .uwc file. This is done by providing the application name as an argument to the .exe. Invoking the .exe will result in a list of valid arguments, one of which: /I can be used to determine the target module the compiler is intended for and for which version.
2. Put the main processor, if one exists, into Bridge Mode so that all incoming and outgoing data is transparently presented at the UART of the module.
3. Send the following commands, terminated by a carriage return character <\r>

```
AT+DEL "myapp" +
```

...wait for the response \n00\r then send (again terminated by <\r>)

```
AT+FOW "myapp"
```

...wait for the response \n00\r

4. Then read up to say 32 bytes (can be more, and limit is specified by the UART receive buffer size in the module) from the .uwc file (which is the output of the cross-compilation step), convert those bytes into a hex string which is double the length, prepend that string with AT+FWRH and then terminate with \r so that you have a command that looks like the following example.

```
AT+FWRH "4F2E574F524C442E5557430110CE211000FB0009000D000A48656C6C6F2057"\r
```

...wait for the response \n00\r

