

**USB4D**

## Contents

<b>USB4D .....</b>	<b>1</b>
Driver specifications .....	4
Configuring the device's communication settings .....	4
Adding a communication driver to your project .....	5
About driver worksheets .....	5
Adding and configuring a Standard Driver Sheet .....	5
Checking the Driver Runtime task .....	9
Troubleshooting .....	10
Revision history .....	11

## USB4D Driver

---

USB4D Driver for Communication with US Digital USB4 Devices (version 1.0, last revised 21 Oct 2011).

The USB4D driver enables communication between the Studio system and remote devices using the Proprietary protocol, according to the specifications discussed in this document.

This document assumes that you have read the "Development Environment" section in the main Studio documentation.

This document also assumes that you are familiar with the Microsoft Windows XP/Vista/7 environment. If you are not familiar with Windows, then we suggest using the **Help and Support** feature (available from the Windows **Start** menu) as you work through this document.

## Driver specifications


---

This section identifies all of the software and hardware components required to implement communication between the USB4D driver in Studio and remote devices using the Proprietary protocol.

### Driver files

The USB4D driver package comprises the following files, which are automatically installed in the `Drv` folder of the Studio application directory:

- `USB4D.DLL`: Compiled driver.
- `USB4D.INI`: Internal driver file. *You must not modify this file.*
- `USB4D.MSG`: Internal driver file defining error messages for the possible error codes. (These error codes are described in detail in the [Troubleshooting](#) section.) *You must not modify this file.*
- `USB4D.PDF`: This document, which provides complete information about using the driver.

 **Note:** You must use a compatible PDF reader to view the `USB4D.PDF` file. You can install Acrobat Reader from the Studio installation CD, or you can download it from [Adobe's website](#).

You can use the USB4D driver on the following operating systems:

- Windows XP, Windows Server 2003
- Windows Vista/7, Windows Server 2008
- Windows Embedded Compact 5.x/6.x

### Device specifications

To establish communication, your target device must meet the following specifications:

- Manufacturer:
- Compatible Equipment: USB4 Encoder Data Acquisition USB Device
- Programmer Software: USB4 drivers and development software provided by US Digital ([usdigital.com/support/software/usb4-software](http://usdigital.com/support/software/usb4-software)). At a minimum, you must acquire the file `USB4.DLL` and copy it to the `Bin` folder in the Studio application directory. If you do not, then you will not be able to add the USB4D driver to your project.

### Network specifications

To establish communication, your device network must meet the following specifications:

- Device Communication Port: USB 2.0
- Physical Protocol: USB 2.0
- Logic Protocol: Proprietary
- Device Runtime Software: None
- Specific PC Board: None
- Cable Wiring Scheme: None

## Configuring the device's communication settings

---

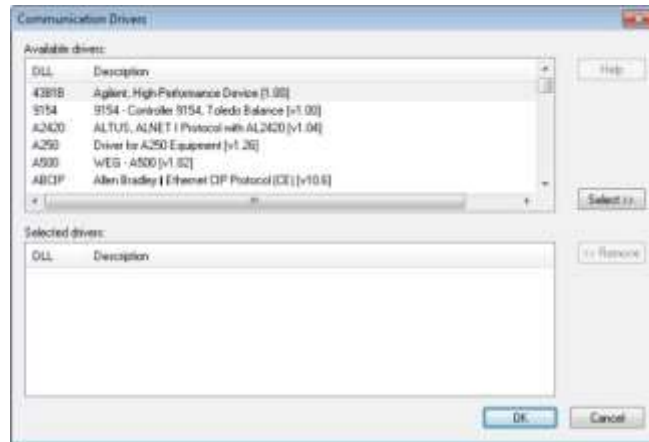
This section explains how to configure the communication settings for the remote device.

All communication settings are handled automatically by the USB4 protocol and development software. You do not need to configure anything on the target device.

## Adding a communication driver to your project

This section explains how to add a communication driver to your project.

1. On the **Insert** tab of the ribbon, in the **Communication** group, click **Add/Remove Driver**. The *Communication Drivers* dialog is displayed.



**Communication Drivers dialog**

2. In the *Available drivers* list, click the communication driver that you want to add.
3. Click **Select**.  
The driver is added to the *Selected drivers* list.
4. Click **OK**.  
The *Communication Drivers* dialog is closed and the selected driver is inserted in the **Drivers** folder in the Project Explorer.

## About driver worksheets

Like the other parts of your project, communication with remote devices is controlled by worksheets. This section explains how to add worksheets to your project and then configure them to associate project tags with device registers.

Each selected driver includes one or more Standard Driver Sheets (SDS). Standard Driver Sheets can be inserted to define tag/register associations that are triggered by specific project behaviors.

The configuration of these worksheets is described in detail in the "Communication" chapter of the *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only USB4D driver-specific parameters and procedures are discussed here.

### **Adding and configuring a Standard Driver Sheet**

By default, a communication driver does not include any Standard Driver Sheets. This section explains how to add a Standard Driver Sheet to your project and then configure it.

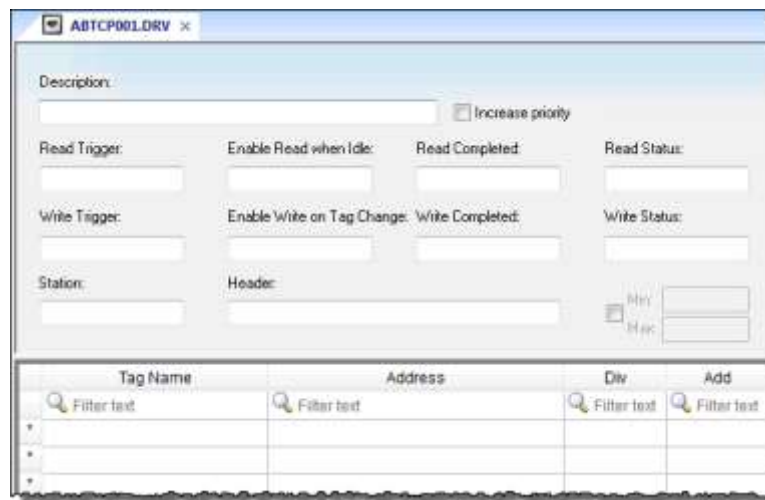
The USB4D driver must be added to the project before you can configure any of its worksheets. For more information, see [Adding a communication driver to your project](#) on page 6.

Standard Driver Sheets can be inserted to define additional tag/register associations that are triggered by specific project behaviors.

**Note:** Most of the settings on this worksheet are standard for all drivers; for more information about configuring these settings, see the “Communication” chapter of the *Technical Reference Manual*. The **Station** and **I/O Address** fields, however, use syntax that is specific to the USB4D driver.

1. Do one of the following.
  - On the **Insert** tab of the ribbon, in the **Communication** group, click **Driver Sheet** and then select **USB4D** from the list.
  - In the **Comm** tab of the Project Explorer, right-click the **USB4D** folder and click **Insert** on the shortcut menu.

A new USB4D driver worksheet is inserted into the **USB4D** folder, and then it is automatically opened for configuring.



**Standard Driver Sheet**

**Note:** Worksheets are numbered in order of creation, so the first worksheet is USB4D001.drv.

2. Configure the Station and Header fields as described below.

**Station**

Specify the ID number of the target USB4 device, between 0 and 3.

You can also specify an indirect tag (e.g. {MyStation}) , but the tag that is referenced must follow the same syntax and contain a valid value.

**Note:** You cannot leave the **Station** field blank.

**Header**

Specify one of the following register types:

**REG**

To access the Incremental Encoder registers.

**DI**

To access the Digital Inputs.

**DO**

To access the Digital Outputs.

**ANALOG**

To access the Analog Outputs.

After you edit the **Header** field, the development application checks the syntax to determine if it is valid. If the syntax is invalid, then the development application automatically inserts a default value of REG.

You can also specify an indirect tag (e.g. {MyHeader}), but the tag that is referenced must follow the same syntax and contain a valid value.

- For each tag/register association that you want to create, insert a row in the worksheet body and then configure the row's fields as described below.

**Tag Name**

Type the name of the project tag.

**Address**

Specify the address of the associated device register.

For REG register type, use the following syntax:

*Register.From\_bit.To\_bit*

...where...

**Register**

The specific register to be read.

**From\_bit**

Read from this bit to the bit specified by *To bit*.

This is an optional parameter. If no bit is specified, then the entire register will be read.

**To\_bit**

Read up to this bit.

This is an optional parameter. If no bit is specified, then up to the first bit (position 0) will be read.

For DI register type, specify one of the eight Digital Input bits (7 through 0). Bit 7 is read/write. Bits 6 through 0 are read only.

For DO register type, specify one of the eight Digital Output bits (7 through 0). All eight bits are read/write.

For ANALOG register type, specify one of the four Analog Output channels (3 through 0).

You can also specify an indirect tag (e.g. {MyAddress}), but the tag that is referenced must follow the same syntax and contain a valid value.



**Note:** Each Standard Driver Sheet can have up to 4096 rows. However, the **Read Trigger**, **Enable Read When Idle**, and **Write Trigger** commands attempt to communicate the entire block of addresses that is configured in the sheet, so if the block of addresses is larger than the maximum block size that is supported by the driver protocol, then you will receive a communication error (e.g., "invalid block size") during run time. Therefore, the maximum block size imposes a practical limit on the number of rows in the sheet.

For examples of how device registers are specified using **Header** and **Address**, see the following table.

**Examples of Header and Address fields in Standard Driver Sheet**

Device Register	Header	Address
All bits of Incremental Encoder register 3	REG	3
Bits 4 thru 0 of Incremental Encoder register 3	REG	3 . 4
Bits 24 thru 4 of Incremental Encoder register 3	REG	3 . 24 . 4
Digital Input bit 6	DI	6
Digital Output bit 3	DO	3
Analog Output channel 2	ANALOG	2

For more information about the device registers and addressing, please consult the manufacturer's documentation.

4. Save and close the worksheet.

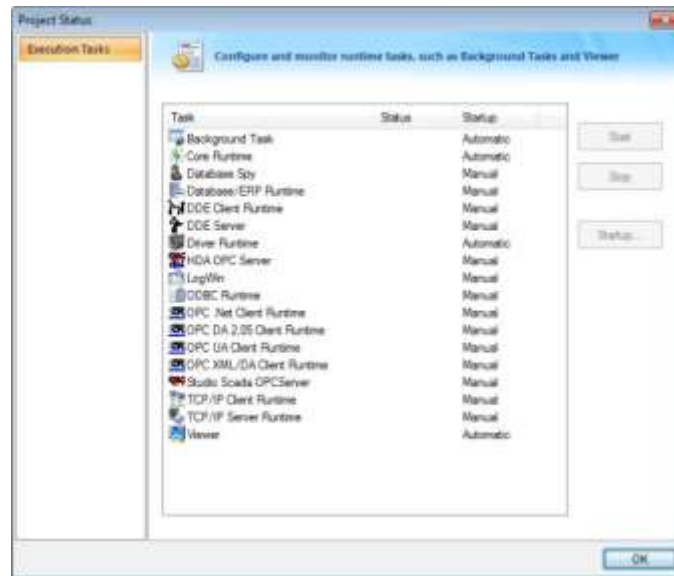


## Checking the Driver Runtime task

This section describes how to check the status of the Driver Runtime task in the list of execution tasks.

The Driver Runtime task handles communication with remote devices and the processing of the driver worksheets. By default, the task is configured to start up automatically when the project is run, but you can check it for yourself.

1. On the **Home** tab of the ribbon, in either the **Local Management** or the **Remote Management** group (depending on where your project server will be running), click **Tasks**.  
The *Project Status* dialog is displayed.



*Project Status dialog*

2. Verify that the **Driver Runtime** task is set to **Automatic**.
  - If the setting is correct, then proceed to the next step.
  - If the **Driver Runtime** task is set to **Manual**, select the task and then click **Startup** to change the task to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.

## Troubleshooting

This section lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.

### Checking status codes

If the USB4D driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** and **Write Status** fields of the driver sheets will receive a status code. Use this status code and the following tables to identify what kind of failure occurred and how it might be resolved. **Status codes for USB4D driver**

Status Code	Error	Possible Causes	Procedure To Solve
0	OK	Communication without problems	None required
1	DEVICE_NOT_OPEN		
2	FAILED_TO_ACQUIRE_MUTEX		
3	FAILED_TO_DOWNLOAD_FIRMWARE		
4	FATAL_ERROR		
5	FIFO_BUFFER_EMPTY		
6	INVALID_A2D_CHANNEL		
7	INVALID_COUNTER_MODE		
8	INVALID_D2A_CHANNEL		
9	INVALID_D2A_MODE		
10	INVALID_DEVICE_NUMBER		
11	INVALID_ENCODER_NUMBER		
12	INVALID_MODULE_NUMBER		
13	INVALID_PARAMETER		
14	INVALID_QUADRATURE_MODE		
15	INVALID_REGISTER_NUMBER		
16	INVALID_SIGNAL_LENGTH_MODE		
17	MODULE_NUMBER_ALREADY_ASSIGNED		
18	MODULE_NUMBER_NOT_FOUND		
19	NO_AVAILABLE_MODULE_ADDRESSES		
20	USB4_INVALID_D2A_VALUE		
30	RX_232_FAILURE		
31	TX_232_FAILURE		
32	NO_DEVICES_FOUND		
33	OLD_FIRMWARE_DETECTED		

## Common status codes

Status Code	Description	Possible Causes	Procedure To Solve
0	OK	Communicating without error.	None required.
-15	Timeout waiting for message to start	<ul style="list-style-type: none"> <li>Disconnected cables.</li> <li>PLC is turned off, in stop mode, or in error mode.</li> <li>Wrong station number.</li> <li>Wrong parity (for serial communication).</li> </ul> Wrong RTS/CTS configuration (for serial communication).	<ul style="list-style-type: none"> <li>Check cable wiring.</li> <li>Check the PLC mode — it must be RUN.</li> <li>Check the station number.</li> <li>Increase the timeout in the driver's advanced settings.</li> </ul> Check the RTS/CTS configuration (for serial communication).

## Monitoring device communications

You can monitor communication status by establishing an event log in Studio's *Output* window (LogWin module). To establish a log for Field Read Commands, Field Write Commands and Serial Communication, right-click in the *Output* window and select the desired options from the pop-up menu.

You can also use the LogWin module to establish an event log on a remote unit that runs Windows Embedded. The log is saved on the unit in the `celog.txt` file, which can be downloaded later.

If you are unable to establish communication between Studio and the target device, then try instead to establish communication using the device's own programming software. Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

## Contacting Technical Support

If you must contact Technical Support, please have the following information ready:

- **Operating System** and **Project Information**: To find this information, click **Support** in the **Help** tab of the ribbon.
- **Driver Version** and **Communication Log**: Displays in the *Output* window (LogWin module) when the driver is enabled and the project is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

## Revision history

This section provides a log of all changes made to the driver.

### Revision history

Driver Version	Revision Date	Description of Changes	Author
1.0	21 Oct 2011	<ul style="list-style-type: none"> <li>Initial release of driver</li> </ul>	André Korbes
1.0	23 Feb 2012	<ul style="list-style-type: none"> <li>Revised driver document for language and usability.</li> </ul>	Michael D. Hayden
1.0	14 May 2012	<ul style="list-style-type: none"> <li>Added note about copying USB4.DLL to the Bin directory.</li> </ul>	Michael D. Hayden