

OWAdapterFlex1.10a - based on OneWireAPI V1.10

One of the cool features of the current 1-Wire adapters is the ability to adjust the 1-Wire waveform "shape". Defined as "flex" settings in the data sheet, the default power-on settings are optimized for typical 1-Wire speeds on a short bus. However, in many weather station applications, you may find yourself forced to use a long bus to connect your weather station. This increases the likelihood 1-Wire errors.

When faced with long cable lengths, many users are opting to use a 1-Wire hub. These hubs simply switch sections of the 1-Wire bus in and out to access the devices on the corresponding leg of the hub. Since there is no "repeater" function (hubs are simply switches), this does not always fix the problem. Instead, adjusting the flex parameters for the longer lengths would minimize errors and reduce the need for 1-Wire hubs.

There are four configurable parameters for 1-Wire communications:

- Pull-Down Slew Rate Control (PDSRC) - limits the slew rate on falling edges (e.g., at the beginning of time slots, to reduce ringing).
- Strong Pull-Up Duration (SPUD) - adjust the time the strong pull-up is active, increasing the recover of the 1-Wire bus to it's high state.
- Write-1 Low Time (W1LT) - extends the Write 1 low time (allows the current flow through the network to end slowly, to prevent voltage spikes from inductive kickback)

Data Sample Offset (DSO) - delays the time point when reading a bit from the 1-Wire bus (gives the network more time to stabilize, to get a higher voltage margin). Setting this parameter also adds extra recovery time between Write 0 time slots (allows more energy transfer through the network, to replenish the parasite power supply of the devices on the bus).

Unfortunately, access to the 1-Wire Flex settings are not available in the standard 1-Wire Java API. The OWAdapterFlex library is a modified version of Dallas Semiconductor's OneWireAPI. It contains added API methods to change the 1-Wire Flex. In addition to the standard API, the following methods are included:

- `public void setPDSRC(int value)`
- `public void setSPUD(int value)`
- `public void setW1LT(int value)`
- `public void setDSO(int value)`

Rather than repeat the information, a detailed description of how these parameters affect your 1-Wire bus is in the DS2480 Serial 1-Wire Line Driver data sheet (the active device in the DS9097U serial-to-1-Wire driver), You can download it at

<http://datasheets.maxim-ic.com/en/ds/DS2480B.pdf>

The OWAdapterFlex library is a drop-in replacement for the existing OneWireAPI. You shouldn't have to make any changes in your code. Once installed, you can access the flex routines when using a DS2480-based 1-Wire adapter such as the DS9097U. Note this driver may work with the DS9490R USB adapter but has not been tested yet.

The following code snippet (from SimpleWeather) shows how you can implement the new methods:

```
...
adapter = OneWireAccessProvider.getAdapter(ADAPTER_TYPE,
                                           ONE_WIRE_SERIAL_PORT);
if (adapter != null)
{
    System.out.println("Found Adapter: " + adapter.getAdapterName() +
                      " on Port " + adapter.getPortName());
}
else
{
    System.out.println("Error: Unable to find 1-Wire adapter!");
    System.exit(1);
}

adapter.setPDSRC(4); // set PDSLC to 1.10 V/uS
adapter.setSPUD(3); // set SPUD to 262 ms
adapter.setW1LT(2); // set W1LT to 10uS
adapter.setDSO(1); // set DSo to 4uS
...
```

Included in the download is the JavaDocs for DSPortAdapter, the root class for all 1_wire adapters. Included in this version is the documentation for the above methods. I've included the relevant section below.

setPDSRC

public void setPDSRC(int rate) throws OneWireIOException, OneWireException

Sets the 1-Wire Pull-Down Slew Rate Control (PDSRC). Valid values are 0 through 7. Default power-on value is 0. See the DS2480 data sheet for further information.

Parameters:

Rate - the selector for the desired slew rate selector in volts/microsecond

- 0 = 15.0 V/uS (nominal)
- 1 = 2.20 V/uS
- 2 = 1.65 V/uS
- 3 = 1.37 V/uS
- 4 = 1.10 V/uS
- 5 = 0.83 V/uS

- 6 = 0.70 V/uS
- 7 = 0.55 V/uS

throws

OneWireIOException on a 1-Wire communication error

OneWireException on a error with the 1-Wire adapter or the adapter does not support this operation

setSPUD

public void setSPUD(int delay) throws OneWireIOException, OneWireException

Sets the 1-Wire Strong Pull-Up Delay (SPUD). Valid values are 0 through 7. Default power-on value is 4. See the DS2480 data sheet for further information.

Parameters:

delay - the selector for the desired delay in milliseconds

- 0 = 16.4 ms
- 1 = 65.5 ms
- 2 = 131 ms
- 3 = 262 ms
- 4 = 524 ms (nominal)
- 5 = 1048 ms
- 6 = dynamic
- 7 = infinite

throws:

OneWireIOException - on a 1-Wire communication error

OneWireException - on a error with the 1-Wire adapter or the adapter does not support this operation

setW1LT

public void setW1LT(int time) throws OneWireIOException, OneWireException

Sets the 1-Wire Write-1 Low Time (W1LT). Valid values are 0 through 7. Default power-on value is 0. See the DS2480 data sheet for further information.

Parameters:

delay - the selector for the desired Write-1 Delay time in microseconds

- 0 = 8 uS (nominal)
- 1 = 9 uS
- 2 = 10 uS
- 3 = 11 uS

- 4 = 12 uS
- 5 = 13 uS
- 6 = 14 uS
- 7 = 15 uS

Throws:

OneWireIOException - on a 1-Wire communication error

OneWireException - on a error with the 1-Wire adapter or the adapter does not support this operation

setDSO

public void setDSO(int offset) throws OneWireIOException, OneWireException

Sets the 1-Wire Data Sample Offset and Write 0 Recovery Time (DSO/W0RT). Valid values are 0 through 7. Default power-on value is 0. See the DS2480 data sheet for further information.

Parameters:

offset - the selector for the desired offset in microseconds

- 0 = 3 uS (nominal)
- 1 = 4 uS
- 2 = 5 us
- 3 = 6 uS
- 4 = 7 uS
- 5 = 8 uS
- 6 = 9 uS
- 7 = 10 uS

throws:

OneWireIOException - on a 1-Wire communication error

OneWireException - on a error with the 1-Wire adapter or the adapter does not support this operation

By implementing this library and optimizing the setting (possibly by experimentation), you can minimize your 1-Wire bus errors, improve noise immunity, and possibly eliminate the need for a 10-Wire hub.