ISL55161-4 and Venus Family Quick Start Application Note

Rev C02: May 20, 2013

This document contains information on a product under development. The parametric information contains target parameters that are subject to change.

Document Revision History

Revision	Date	Description
A01	10/20/2005	Initial Draft
B01	11/10/2005	Added PMU FI mode example
B02	06/05/2006	Applicable to all Venus Family products: Venus, VenusPlus, Venus2, etc Expanded FV to show remote sense and MI
C01	06/18/2012	Added ISL55161/2/3/4
C02	05/20/2013	Changed to Elevate Format

Table of Contents

1 Introduction

This document describes the steps to perform an initial system check-out of the ISL55161-4 and Venus Family devices. The Venus Family includes Venus, and Venus3 and Venus4. Please refer to the datasheets for any specific differences. In this document, the ISL55161-4 and all Venus products will be referred to as a part of the Venus Family.

These instructions assume the customer system can set the Power supplies, provide a PLL Clock (if needed), has a mechanism to read/write registers, has the ability to measure voltages (either using an external DMM or system resource), provide a simple DATA stream, and so on.

Important Note: The steps described below illustrate a cause and effect to demonstrate how to interface with the device. In a normal application; sequencing, calibration and other factors may require the registers to be written in a different order. Please refer to the Planet ATE Software Driver documentation for details.

1.1 Set Power Supplies

The first step is to apply the appropriate voltage. After power is applied, it is required to toggle the hardware reset (active high) on Venus Family device or issue the Software Reset since there is no Power-On Reset (POR).

Verify the voltage at the Venus Family pins or within close proximity to ensure there are no IR drops. In addition, the customer should verify the approximate current flow for each supply.

Note: Refer to the appropriate datasheet for typical quiescent current results.

Table 1: Venus (Classic) Power-ON Current Values

Supply	Voltage (V)	Approximate Current (mA)	
		PLL_CK = 50MHz	PLL_CK = 125MHz
VCC_SV	13.0	30	30
VCC	8.0	30	30
VEE	-3.0	-60	-60
VDD	3.3	200	240
VREF	3.0	0	0
VOH	3.0	0	0
VOL	0.0	0	0

Notes:
1) The VOH and VOL should have minimal current since the COMP pins are in a static state and typically drive a high impedance input.
2) In most systems, it may not be possible to measure the supply currents.

1.1.1 Measure REXT and CAP_PLL Voltages

The next step is to measure voltage of the REXT and CAP_PLL pins.

REXT voltage should track the VREF voltage within 25mV.

The CAP_PLL voltage is a function of the PLL_CK. In addition, the CAP_PLL voltage is dependant on the Vswing and Fclamp registers. Table 2 lists the CAP_PLL vs PLL_CK assuming the power-ON default Vswing (00b) and Fclamp (00b) settings.

Table 2: Default CAP_PLL Voltages

PLL_CK (MHz)	CAP_PLL (V)
50	1.0
75	1.2
100	1.5
125	1.7

1.2 Register/RAM Access (CPU Port Transactions)

The following steps perform simple register/RAM access (write/read) to verify to CPU port is functioning.

1.2.1 Read Die ID Register

Read the Die ID registers (address = 0x00FF); Table 3 lists the expected return values for different silicon revisions. Some of the revisions may have changed since the writing of this document.

Table 3: Die ID Register Values

Venus Read-back Data (HEX)	Venus3 Read-back Data (HEX)	Venus4 Read-back Data (HEX)
0x0647 (Rev 7)	0x0677 (Rev 7)	0x0685 (Rev 5)
0x064# (Rev #)	0x067# (Rev #)	0x068# (Rev #)

ISL55161 Read-back Data (HEX)	ISL55162 Read-back Data (HEX)	ISL55163 Read-back Data (HEX)	ISL55164 Read-back Data (HEX)
0x0686 (Rev 6)	0x0678 (Rev 8)	0x0686(Rev 6)	0x064C (Rev C)
0x068# (Rev #)	0x067# (Rev #)	0x068# (Rev #)	0x064# (Rev #)

1.2.2 Register Access

Follow Table 4 to verify the basic ability to write/read registers. The following bullet items highlight some key aspects of a register access:

- Address bit A7 is the channel decoder
  - Channel 0 is located between address 0 to 111
  - Channel 1 is located between address 128 to 239
- Central Registers are located between address 112 to 127 (within the Channel 0 address space)
- WE bit must be ‘1’ for the corresponding Data bits to be written
- WE bits (typically) read back ‘0’
- Writing to Undefined Data bits has no effect
- Undefined Data bits (typically) read back ‘0’
- A RESET sets all registers to a default value of ‘0000’ (except Read-Only registers)
- Refer to the CPU Protocol Timing diagram for details

Table 4: Write/Read Registers

Description	Address (HEX)	Write Data (HEX)	Read Data (HEX)
Ch#0: Set Driver/En Term = 50 Ohm	0x0040	0x003F	0x001B
Ch#0: Clear EN Term. DATA Term should be unchanged.	0x0040	0x0020	0x0003
Ch#1: Enable COMPA Coarse/Fine Delay to full code	0x00D1	0xFFFF	0x5F4F
Ch#1: Disable COMPA Fine Delay. Leave Coarse delay unchanged	0x00D1	0x0080	0x5F0F
Parallel Write: Set DVL Radj (need to set ‘Parallel Load’ bit)	0x0047	0x001F	0x000F (read on all chans)
Central Reg: Enable MONITOR on Channel 0	0x0070	0x000F	0x0005
Central Reg: Software RESET	0x0073	<don’t care>	0x0000 (all registers)

1.2.3 RAM Access

Follow Table 4 to verify the basic ability to write/read RAM. The following bullet items highlight some key aspects of a RAM access:

- Address bit A7 is the channel decoder
  - Channel 0 is located between address 0 to 111
  - Channel 1 is located between address 128 to 239
- Central Registers are located between address 112 to 127 (within the Channel 0 address space)
- The Read-back Data matches the Written Data
- A RESET does NOT set the RAM contents to a default value. They are left unchanged
- A RAM read-back requires 2 addition clock cycles
- Refer to the CPU Protocol Timing diagram for details

Table 5: Write/Read Registers

Description	Address (HEX)	Data (HEX)
Ch#0: DVH Value DAC	0x0000	0xFFFF
Ch#0: DVH Offset DAC	0x0010	0xAAAA
Ch#0: DVH Gain DAC	0x0020	0x5555
Ch#1: CVA Value DAC	0x0083	0x1234
Parallel Write: DVL Offset DAC (need to set 'Parallel Load' bit)	0x0011	0xABCD (read on all chans)

1.3 Reset State

Whenever a Hardware or Software reset is issued, the device is configured into the following state.

Note: The rest of the examples rely on the default (reset) configuration to demonstrate the minimum register transactions to achieve the desired configuration.

- RAM – Unknown at power ON. Left unchanged if reset issued from a valid configuration
- Registers – all registers are initialized to '0'; which implies:
  - Driver Block
    - Driver is in high-Z mode
    - Data/En sourced from CPU control
    - Deskews disabled
    - All switches open
    - Data/En Input Termination none
    - Voltage Range 0
  - Comparator Block
    - All switches open
    - COMP pins sourced from Real-Time comparator.
    - Deskews disabled
    - Voltage Range 0
  - PMU Block
    - PMU Disabled
    - All switches open
    - Tight loop
    - All IR open
    - Voltage Range 0
    - V-Clamps are disabled
    - Monitor is in high-Z
  - PLL
    - No input termination
    - Vswing/Fclamp set to '0'.

1.4 Set Driver (DVH/DVL) Level

The following steps are used to output a Driver voltage at DOUT_#. The CPU-Data/En register control bits are used to set the driver state: Set CPU-Data high for DVH and low for DVL.

In addition to measuring the voltage at the DOUT_# pin, the customer should (could) also confirm the voltage at the DVH_# and DVL_# pin (driver compensation capacitors).

Note: The 'Output Voltage' is the approximate voltage since these are un-calibrated settings.

Table 6: Set Driver DVH/DVL Levels

Description	Address (HEX)	Data (HEX)	Output Voltage
Software Reset	0x0073	<don't care>	High-Z
Ch#0: Set DVH = 3.0V (Driver defaults to VR0)	0x0000	0xDFFF	High-Z
Ch#0: Set DVH Offset = 0.0	0x0010	0x7FFF	High-Z
Ch#0: Set DVH Gain = 1.0	0x0020	0x7FFF	High-Z
Ch#0: Set DVL = 0.0V	0x0001	0x1FFF	High-Z
Ch#0: Set DVL Offset = 0.0	0x0011	0x7FFF	High-Z
Ch#0: Set DVL Gain = 1.0	0x0021	0x7FFF	High-Z
Ch#0: Enable Driver (CPU-En = 1) and set Driver High (Data = 1) Found in Central Register space	0x0071	0x000F	3.0 (DVH)
Ch#0: Set DVH = 1.5V	0x0000	0x7FFF	1.5
Ch#0: Set DVH = 3.5V	0x0000	0xFFFF	3.5
Ch#0: Set Voltage Range 1 (VR1)	0x0064	0x0005	7.0 (RAM contents don't change)
Ch#0: Set Driver Low (Data = 0)	0x0071	0x0002	0.0 (DVL)
Ch#0: Set DVL = -1.0V Driver is already in VR1	0x0001	0x0000	-1.0
Ch#0: Set Voltage Range 0 (VR0)	0x0064	0x0004	-0.5 (RAM contents don't change)
Ch#0: Set DVL = 0.0V	0x0001	0x1FFF	0.0

Note: Repeat the above steps for other channels.

1.5 Ring Oscillator (Driver/Comparator Outputs)

The following steps use the internal Ring Oscillator to generate an approximate 15 MHz pulse. The output waveforms can be monitored at the DOUT_# and COMPA/B_# pins. The Ring Oscillator is a useful debug feature since the customer doesn't need to provide a DATA_# stream to the device. The Ring Oscillator is used in production testing, demonstrations and for initial troubleshooting. This feature is typically not useful in a customer system application.

Table 7: Ring Oscillator Output Waveforms

Description	Address (HEX)	Data (HEX)	DOUT_# Output	COMPA/B_# Output
Software Reset	0x0073	<don't care>	High-Z	Unknown (internal node floats)
Set Fclamp/Vswing/PLL Term based on PLL_CK	0x0072	See Datasheet	High-Z	Unknown (internal node floats)
Ch#0: Set DVH = 3.0V (Driver defaults to VR0)	0x0000	0xDFFF	High-Z	Unknown (internal node floats)
Ch#0: Set DVH Offset = 0.0	0x0010	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set DVH Gain = 1.0	0x0020	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set DVL = 0.0V	0x0001	0x1FFF	High-Z	Unknown (internal node floats)
Ch#0: Set DVL Offset = 0.0	0x0011	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set DVL Gain = 1.0	0x0021	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVA = 1.5V (Comparator defaults to VR0)	0x0003	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVA Offset = 0.0	0x0013	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVA Gain = 1.0	0x0023	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVB = 1.5V (Comparator defaults to VR0)	0x0004	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVB Offset = 0.0	0x0014	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVB Gain = 1.0	0x0024	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: set Driver = Real-Time Data	0x0042	0x0003	Matches DATA_0 input	Matches DOUT_0 state
Ch#0: Enable Driver (CPU-En = 1) Found in Central space	0x0071	0x000C	Matches DATA_0 input	Matches DOUT_0 state
Ch#0: Connect and Enable Ring Oscillator to DATA path	0x0055	0x0061	Most likely Low	Matches DOUT_0 state
Ch#0: Trigger Ring Oscillator	0x0055	0x0180	15MHz 3V swing	15MHz VOH/VOL swing

1.5.1 Real-Time Data

To demonstrate a real-time data pattern using the DATA_# to DOUT_# path; the customer should skip the last 2 steps (don't enable the Ring Oscillator). The customer needs to provide a valid DATA_# input signal. The Venus device is configured for CPU-EN = High (Driver enabled).

1.6 Monitor Output

The following steps use the internal Test & Cal diagnostic mux to output a known voltage at the MONITOR pin.

Note: The 'Output Voltage' is the approximate voltage since these are un-calibrated settings.

Table 8: Monitor Output

Description	Address (HEX)	Data (HEX)	MONITOR Output
Software Reset	0x0073	<don't care>	High-Z
Ch#0: Set VFV = 3.5V (PMU defaults to VR0)	0x0007	0xFFFF	High-Z
Ch#0: Set VFV Offset = 0.0V	0x0017	0x7FFF	High-Z
Ch#0: Set VFV Gain = 1.0V	0x0017	0x7FFF	High-Z
Ch#0: Set Sel-Mu-Diag to source from Test & Cal mux and Diag to source from VFV DAC	0x0063	0x00D4	High-Z
Central: Enable Monitor from Ch#0	0x0070	0x0003	3.5

1.6.1 Measure Junction Temperature

The following steps show how to measure and calculate the junction temperature using the internal temperature sensors.

Tj = (Va – Vb) * 1637 – 221

The measured temperature will be dependant on if there is a heat sink and/or air-flow present.

Table 9: Measure Junction Temperature

Description	Address (HEX)	Data (HEX)	MONITOR Output
Software Reset	0x0073	<don't care>	High-Z
Ch#0: Set Sel-Mu-Diag to source from Test & Cal mux and Diag to source from PMU-Va-Tj	0x0063	0x00D6	High-Z
Central: Enable Monitor	0x0070	0x0003	Measure Va
Ch#0: Set Test & Cal to PMU-Vb-Tj	0x0063	0x0017	Measure Vb (apply above formula)

1.7 PMU Force Voltage (FV) Mode

The following steps connect the Channel #0 PMU to the DOUT_0.

Note: The 'Output Voltage' is the approximate voltage since these are un-calibrated settings.

Table 10: PMU Force Voltage (FV) Mode

Description	Address (HEX)	Data (HEX)	DOUT_# Output	MONITOR Output
Software Reset	0x0073	<don't care>	High-Z	High-Z
Ch#0: Set VFV = 3.0V (PMU defaults to VR0)	0x0007	0xDFFF	High-Z	High-Z
Ch#0: Set VFV Offset = 0.0	0x0017	0x7FFF	High-Z	High-Z
Ch#0: Set VFV Gain = 1.0	0x0027	0x7FFF	High-Z	High-Z
Ch#0: Set IR=8mA	0x0061	0x0140	High-Z	High-Z
Ch#0: Enable PMU (default is FV mode)	0x0060	0x3000	High-Z	High-Z
Central: Enable Monitor to Ch#0 (default is MV mode)	0x0070	0x000B	High-Z	Unknown (tracks DOUT)
Ch#0: Connect PMU SV switch to DOUT_0 (central register)	0x0071	0x0030	3.0V	3.0V
Connect 500 Ohm resistor from DOUT to GND. (1)	N/A	N/A	2.22V	2.22V
Ch#0: Measure MI	0x0060	0x000C	2.22V	0.555V (2) (4.4mA)
Ch#0: Sense at DOUT (Loop=1) (default was Tight loop)	0x0060	0x0030	3.0V	0.75V (2) (6mA)

Notes:

1) IR=8mA has Rsense=125 ohm + 50 ohm switch; therefore Vdout = Vprogram * 500 / (500 + 175)
2) MI voltage has +/-1V = +/-Imax relationship. Current = Vdout / 500

1.8 PMU Force Current (FI) Mode

The following steps connect the Channel #0 PMU DOUT_0. A load needs to be connected to the DOUT_0. If no load is present then the FI will cause the PMU to output a voltage at one of the supply rails. The PMU is connected to the DOUT_0 prior to switching into FI mode to prevent a large glitch.

Note: if the part is put into FI mode before connecting to the DOUT pin then the internal voltage will go to one of the supply rails, therefore when the PMU is connected it will cause a glitch on DOUT#.

Note: The 'Output Current' is the approximate current since these are un-calibrated settings.

Table 11: PMU Force Current (FI) Mode

Description	Address (HEX)	Data (HEX)	DOUT_# Output
Software Reset	0x0073	<don't care>	High-Z
Ch#0: Set VFI = 1.0V (+Imax)	0x0008	0xFFFF	High-Z
Ch#0: Set VFI Offset = 0.0	0x0018	0x7FFF	High-Z
Ch#0: Set VFI Gain = 1.0	0x0028	0x7FFF	High-Z
Ch#0: Set IR=2mA	0x0061	0x0120	High-Z
Ch#0: Connect PMU SV switch to DOUT_0 (central register)	0x0071	0x0030	0.0V (FV Mode)
Ch#0: Enable PMU, set FI mode, and set Tight=1 (sense across Rsense instrument amp)	0x0060	0x3033	FI Mode (voltage depends on load)

1.9 Active Load Mode (Venus4, ISL55161, and ISL55163 only)

The following steps explain how to connect the active diode bridge load for Channel #0 to source current. An external load needs to be connected to the DOUT_0, if no external load is present then no current will sink or source. The external load must be at least one diode drop below Vtt for current to source. DOUT_0 was connected to a 0V load after completing the software reset in this example case.

Table 12: Active Load setup

Description	Address (HEX)	Data (HEX)	DOUT0
Software Reset	0x0073	<don't care>	High-Z
Ch#0: Set Vmid =VR0	0x0064	0x4000	High-Z
Ch#0: Set Vtt =1.5V	0x0002	0x7FFF	High-Z
Ch#0: Set Vtt offset =0	0x0012	0x7FFF	High-Z
Ch#0: Set Vtt gain =1	0x0022	0x7FFF	High-Z
Ch#0: Set Sink/Source-Adj value =nominal	0x0065	0x0088	High-Z
Ch#0: Set I-source value =24mA	0x000B	0xC000	High-Z
Ch#0: Set I-sink value =24mA	0x000C	0xC000	High-Z
Ch#0: Set I-source Dac offset =0	0x001B	0x7FFF	High-Z
Ch#0: Set I-sink Dac offset =0	0x001C	0x7FFF	High-Z

Ch#0: Set I-source Dac gain =1	0x002B	0x7FFF	High-Z
Ch#0: Set I-sink Dac gain =1	0x002C	0x7FFF	High-Z
Ch#0: Set CPU-En0 = 0	0x0071	0x0008	High-Z
Ch#0: active load connect and source/sink enable	0x0042	0x03C0	Connected to Gnd measured 24mA sourcing

1.10 Planet ATE Software Driver and Documentation

At this point, the customer has demonstrated the basic ability to configure the device, set some levels, make some measurements, and output a driver waveform. This should imply the Venus Family device has been properly designed and assembled in the customer system.

The next step is up to the customer to write software drivers which would do the following:

- Configure the device for different modes
- Perform DC calibration
- Set Levels as a function of voltage rather than HEX codes
- Set Deskews
- Configure the PMU for different modes; including proper sequencing
- And more

