



I²C ADDRESSING AND FAULT MONITORING USING THE ACS71020 POWER MONITORING IC DIGITAL INPUT OUTPUT PINS

By Maxwell McNally
Allegro MicroSystems

This application note describes methods of setting the I²C slave address of the ACS71020 and the added functionality of the digital input output (DIO) pins when operating in I²C mode.

INTRODUCTION

The ACS71020 Power Monitoring IC is capable of communicating in either I²C or SPI mode over pins 9 through 12. SPI communication requires four pins but I²C only requires two. If the ACS71020 is configured to output over the SDA and SCL lines for I²C communication, the MOSI (10) and CS (9) pins are freed to take on other functions. Figure 1 shows the pinout of the ACS71020 and an example I²C application circuit.

To facilitate having multiple ACS71020 devices on the same board, DIO_0 and DIO_1 are polled at startup and the supplied voltages are used to calculate an I²C address. If different voltages are applied to each of the ACS71020 devices, their addresses will be set independently at startup by the surrounding hardware and not in software, ensuring repeatable addressing without requiring an additional programming step.

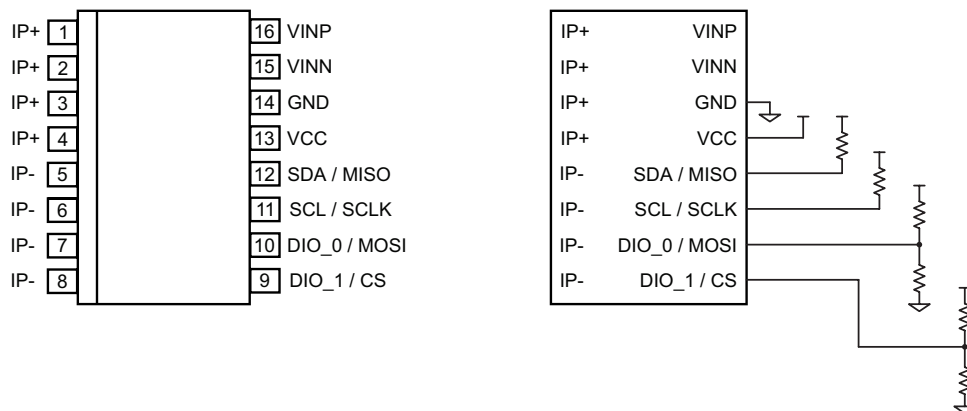


Figure 1. Pinout and I²C Application Circuit Diagram for the ACS71020

After startup, the DIO pins take on another role as push-pull fault outputs. Programmable EEPROM settings allow for different combinations of voltage and current monitoring signals to be routed through the DIO pins. The push-pull outputs pull low when the corresponding operating condition occurs.

I²C Slave Addressing

The ACS71020 supports I²C communication over the SCL and SDA lines at speeds of up to 400 kHz. When the device first powers on, it measures the voltage level on the two DIO pins. It converts both voltage levels into a 4-bit code for a total of sixteen slave addresses. Table 17 in the datasheet and Table 1 below show the sixteen possible I²C configurations that can be set with externally applied voltage. If both pins are pulled to V_{CC}, then the internal slave address stored in EEPROM is used. By default, the value of `i2c_slv_addr` is programmed at the Allegro factory to 127, but this can be changed with programming by the customer.

Table 1. DIO Startup Voltage Addressing

DIO_1	DIO_2	A6	A5	A4	A3	A2	A1	A0	Slave Address (decimal)	
0	0	0	0	1	1	0	0	0	0	97
0	0	0	1	1	1	0	0	0	0	97
0	0	1	0	1	1	0	0	0	1	98
0	0	1	1	1	1	0	0	0	1	99
0	1	0	0	1	1	0	0	1	0	100
0	1	0	1	1	1	0	0	1	0	101
0	1	1	0	1	1	0	0	1	1	102
0	1	1	1	1	1	0	0	1	1	103
1	0	0	0	1	1	0	1	0	0	104
1	0	0	1	1	1	0	1	0	0	105
1	0	1	0	1	1	0	1	0	1	106
1	0	1	1	1	1	0	1	0	1	107
1	1	0	0	1	1	0	1	1	0	108
1	1	0	1	1	1	0	1	1	0	109
1	1	1	0	1	1	0	1	1	1	110
1	1	1	1	EE	EE	EE	EE	EE	EE	EEPROM value

If for any reason the external slave address setting feature is not desired, the DIO polling can be disabled by setting the `i2c_dis_slv_addr`. When this bit is set, the ACS71020 will automatically use the number stored in `i2c_slv_addr` as the I²C slave address regardless of the voltage on the DIO pins. Note that the device must be repowered for these changes to take effect.

Multiple I²C Device Addressing

Bus contention can occur when multiple devices attempt to communicate with the same I²C address. By setting the I²C slave address with external circuitry, up to sixteen ACS71020 devices can be connected to the same bus without the need for independent preprogramming. Table 2

shows the voltage thresholds the ACS71020 uses to determine the code for each DIO pin. These can be cross-referenced with Table 1 to determine the exact I²C slave address knowing the external voltages on the DIO pins.

Table 2. DIO Input Range and Corresponding Code

Voltage Input Range		DIO Code	
0% V _{CC}	25% V _{CC}	0	0
25% V _{CC}	50% V _{CC}	0	1
50% V _{CC}	75% V _{CC}	1	0
75% V _{CC}	100% V _{CC}	1	1

At startup, the ACS71020 will poll the applied voltage set by the external resistors. After the voltage is measured, the push-pull output will take over and drive the output high or low depending on application conditions. Table 3 lists recommended resistor values for the four DIO code levels and the associated high and low voltage levels seen by the microprocessor. Lower resistance values than those listed in Table 3 can impact the ACS71020's ability to drive the DIO voltage to the desired level.

Table 3. Recommended Resistor Values for DIO Pins

Pull-Up Resistor	Pull-Down Resistor	High Voltage	Low Voltage	DIO Code	
None	10 kΩ	3.235 V	2 mV	0	0
10 kΩ	5.1 kΩ	3.183 V	45 mV	0	1
5.1 kΩ	10 kΩ	3.236 V	87 mV	1	0
10 kΩ	None	3.291 V	5 mV	1	1

DIO Output Options

After startup, the DIO pins become diagnostic outputs. There are five internal flags that can be routed through the two DIO pins while communicating in I²C mode. While these flags are always available in volatile memory, the push-pull outputs can be routed to interrupt pins for faster processing of potentially dangerous operating conditions. Refer to the DIO Output Selection section below for information on how to configure the DIO outputs to specific flag combinations.

Voltage Zero Crossing

The ACS71020 is always monitoring the voltage channel to check for a change in sign of the voltage. It uses the zero crossings to trigger calculation of voltage and current RMS, averaging, and power information every half cycle. Every rising zero crossing, the ACS71020 sets the value of `vzerocrossout` to 1 for 32 μs. The delay between the zero crossing and the `vzerocrossout` being set is a consistent 350 μs. One of the output options of `DIO_0` is to pull low

while `vzerocrossout` is set. The length of the pulse can be increased to 256 μ s by setting the value of `delaycnt_sel` to 1. Figure 2 shows DIO_0 pulling low on every rising voltage zero crossing.

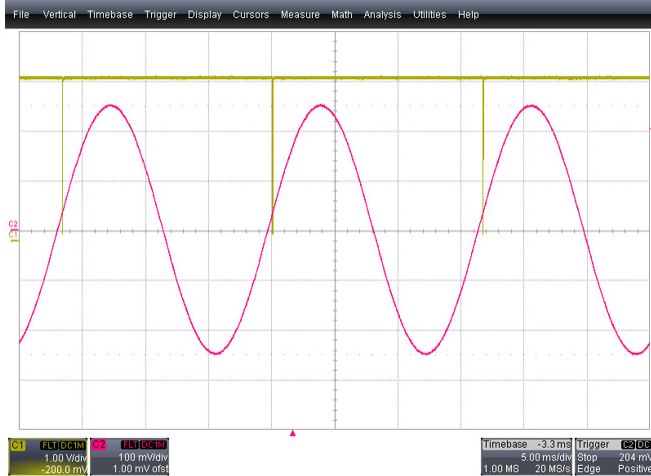


Figure 2. Default Voltage Zero Crossing Output Functionality

The ACS71020 can set `vzerocrossout` after every positive and negative going voltage crossing by writing a 1 to the `halfcycle_en` register. Figure 3 shows the output of DIO_0 in this condition.

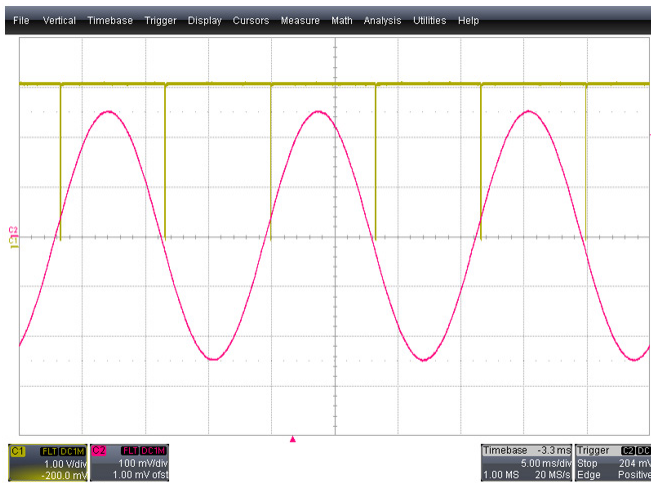


Figure 3. Voltage Zero Crossing Output with `halfcycle_en` Set

A third option is available for the voltage zero crossing output in which the DIO_0 output toggles on every voltage zero crossing, outputting a square wave. Setting the `squarewave_en` bit enables the functionality shown in Figure 4. The square wave always toggles every zero crossing and does not change with the `halfcycle_en` bit.

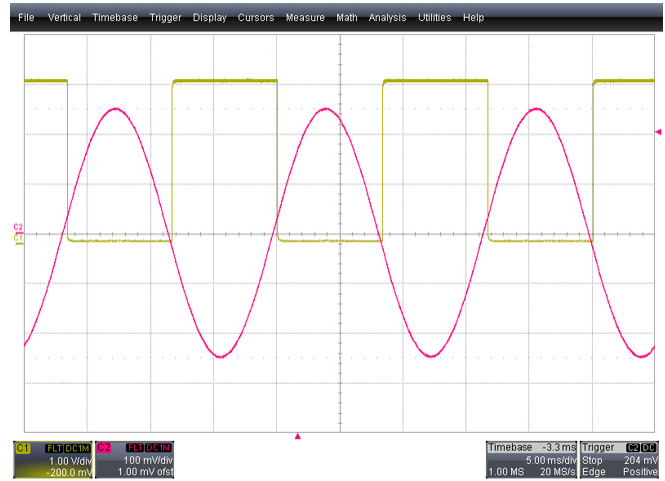


Figure 4. Voltage Zero Crossing Output with `squarewave_en` Set

Overvoltage VRMS

The ACS71020 calculates the voltage RMS after every voltage zero crossing and stores it in the `vrms` register. This can be read over the I²C interface to be stored in the microprocessor memory. If `vrms` ever exceeds the threshold set in the `overvreg` field, the ACS71020 sets the overvoltage flag in volatile memory. Not only can the overvoltage register be read over I²C, the DIO pins can be programmed to pull low when an overvoltage RMS event occurs.

For every code set in the `overvreg` field, the overvoltage trip level is set to 520.127 codes of the `vrms` field due to the difference in register sizes. The trip point range spans the entire `vrms` range. If code 30 was programmed into the `overvreg` field, the overvoltage flag would not be set until `vrms` exceeded code 15603. Figure 5 shows DIO_0 pull low when the RMS voltage of a half cycle increases past the fault threshold.

Setting `vevent_cycs` greater than 1 will hold the overvoltage flag from being set until the fault condition persists for that many extra half cycles. A `vevent_cycs` value of 2 corresponds to 3 overvoltage half cycles that `vrms` must be above the fault threshold, as shown in Figure 6.

Undervoltage VRMS

The ACS71020 undervoltage flag operates in a similar manner to the overvoltage flag. It shares the `vevent_cycs` register with overvoltage detection but has a separate threshold setting for the `vrms` level in `undervreg`. When the value in the `vrms` register falls below the threshold set in `undervreg`, the DIO pin will pull low as shown in Figure 7.

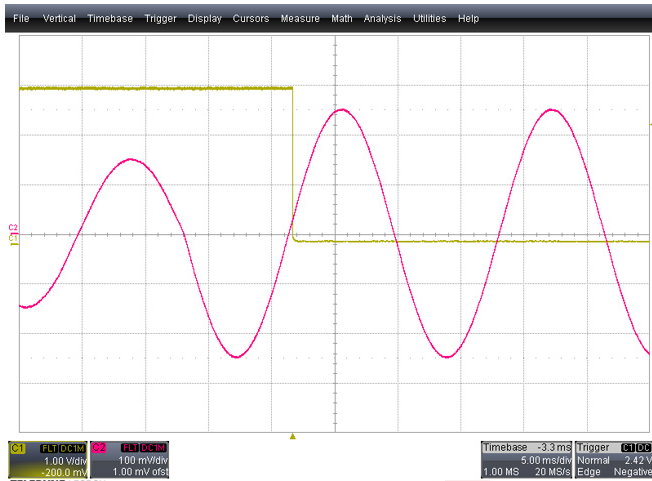


Figure 5. Overvoltage VRMS Detection after 1 Half Cycle



Figure 6. Overvoltage VRMS detection after 3 Half Cycle

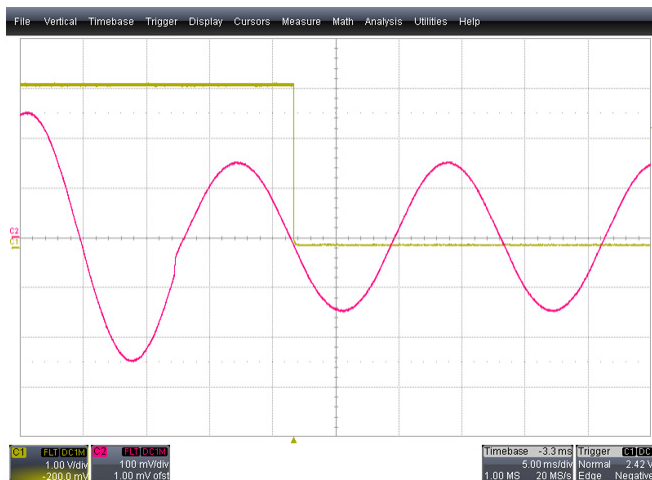


Figure 7. Undervoltage VRMS Detection After 1 Half Cycle

Undervoltage or Overvoltage VRMS

DIO_0 has the option of pulling low in the event of an undervoltage or overvoltage VRMS condition. DIO_0 will be held high for as long as the voltage is within the range specified by the overreg and undervreg. Figure 8 shows the DIO_0 pin pulling low in the case of both under and over VRMS voltage levels, but releasing during intermediate voltage operation.

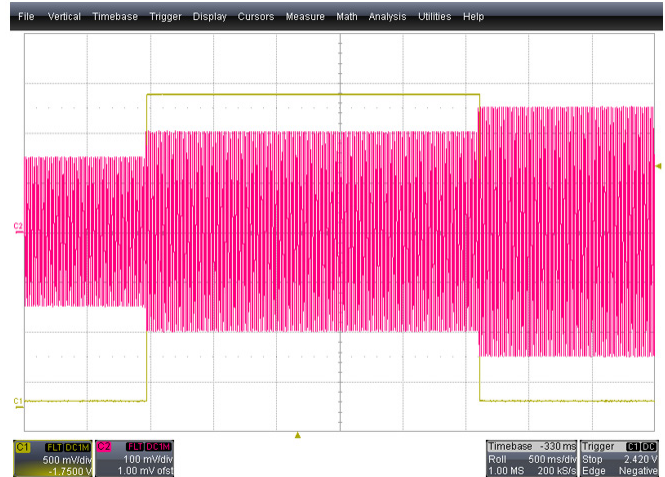


Figure 8. Overvoltage or Undervoltage Flag Detection

Overcurrent Fault

The ACS71020 constantly updates the value of icodes after every sampling event and calculates irms after every voltage zero crossing. Short circuit overcurrent events can be dangerous and the ACS71020 sets the value of faultout to 1 should the instantaneous current in icodes exceed the value set in the fault register. Figure 9 shows the ACS71020 responding to an input current pulse and indicating an overcurrent fault by pulling DIO_1 low. The ACS71020 can report the fault event immediately or delay the response in accordance with the value in fldly. This delay does not include the nominal 5 μ s propagation delay of the fault circuitry. Table 11 in the ACS71020 datasheet shows the delay times available for the overcurrent fault, shown below as Table 4.

The faultlatched register will stay true after the current level has fallen below the fault threshold, while the fault register will not. Writing a 1 to the faultlatched register will clear the bit if the fault condition is no longer present.

Table 4. *ftdly* Selection

Range	Value	Units
0	0	µs
1	0	µs
2	4.75	µs
3	9.25	µs
4	13.75	µs
5	18.5	µs
6	23.25	µs
7	27.75	µs



Figure 9. *DIO_1* Overcurrent Fault Response with *ftdly* 0

DIO Output Selection

Each of the previously mentioned diagnostic flags are available for routing to the outputs of DIO_0 and DIO_1 in different combinations. With only two output pins and four flags, they cannot all be simultaneously accessed through the DIO pins. If the corresponding DIO_0_sel or DIO_1_sel is programmed to 3, multiple fault conditions are or'd together and if any of those conditions are present, the corresponding DIO pin will pull low. Table 5 lists all the available routing options for both DIO pins according to the value programmed in the corresponding DIO select register. Once the microprocessor knows that a fault has occurred, the volatile registers can be read to discern which fault event is occurring and the system can take the appropriate action.

Table 5. *DIO* Flag Routing Options

DIO_Sel	DIO_0	DIO_1
0	Voltage Zero Crossing	Overcurrent Fault
1	Overvoltage RMS	Undervoltage RMS
2	Undervoltage RMS	Overvoltage RMS
3	Overvoltage or Undervoltage RMS	Undervoltage, Overvoltage, or Overcurrent Latched Fault

Conclusion

The ACS71020 is a highly configurable power monitoring chip with the ability to report unexpected or potentially dangerous operating conditions through the push-pull DIO pins. When the corresponding operating condition is met, the DIO pins pull low, easily allowing the system controller to run a corrective action in response to an interrupt. In order to prevent I2C address conflicts, the ACS71020 can set its address at startup based only on circuitry external to the device, preventing the hassle of preprogramming.

Revision History

Number	Date	Description	Responsibility
–	May 28, 2019	Initial release	M. McNally

Copyright 2019, Allegro MicroSystems.

The information contained in this document does not constitute any representation, warranty, assurance, guaranty, or inducement by Allegro to the customer with respect to the subject matter of this document. The information being provided does not guarantee that a process based on this information will be reliable, or that Allegro has explored all of the possible failure modes. It is the customer's responsibility to do sufficient qualification testing of the final product to insure that it is reliable and meets all design requirements.

Copies of this document are considered uncontrolled documents.