a2Spark

An easily customizable I2C analog device

Created by Alexandre Boni
Designed for Digispark

« *a2Spark is an I2C slave device with two 10-bits analog inputs and two digital alarm outputs. a2Spark is designed to run on Digispark, a tiny Arduino controller based on Atmel ATTiny85 controller.* »

Features

- I2C Slave Device
  - Default address: 0x42
- 5V and 3.3V compatible
- 10-bit ADC
  - 2 Channels
    - Configurable Low-Pass filter
  - 2 Alarms
    - Schmitt trigger (hysteresis)
    - Dedicated hardware outputs
    - Edge or level mode
- Internal Temperature Sensor
  - 10-bit resolution
  - Configurable offset temperature
  - Unit selection
    - Kelvin
    - Celsius
    - Fahrenheit
- Powered by the Atmel ATTiny85

Contents

History

Release 1.2

- Add reverse feature in Alarm Mode
  - 3 new modes added
  - Affected registers: AL1CR (ALM2:0, ALME)
- Add Low-Pass filter
  - Weighted average feature removed
  - AVGnR renamed LPFnR, AVEN renamed LPEN, AG renamed LF
- Add Offset temperature feature in TCR
- Firmware version 8

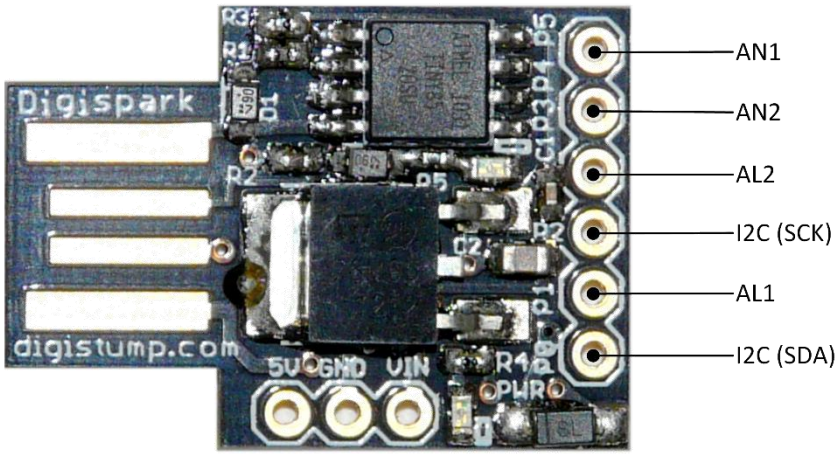
Release 1.1

- Add internal temperatures feature.
- Affected registers: TCR, TDRL, TDRH
- Firmware version 7

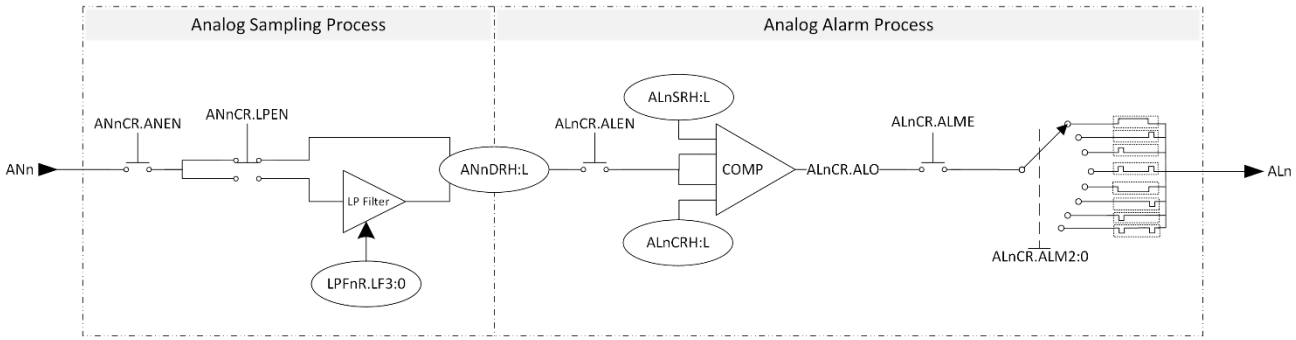
Release 1.0

- Initial release of the datasheet
- Firmware version 6

I. Hardware Overview



II. Software Overview



Caption:

- n = 1 or 2
- ANn is an analog input, while ALn is a digital output.

III. Registers Summary

Name	**ID**	**Offset**	**Access**
I2C Address Register	ADDR	0x00	RO
Firmware Version Register	FWVR	0x01	RO
Analog1 Configuration Register	AN1CR	0x04	RW
Low-Pass Filter1 Register	LPF1R	0x05	RW
Analog1 Data Register Low	AN1DRL	0x06	RO
Analog1 Data Register High	AN1DRH	0x07	RO
Alarm1 Configuration Register	AL1CR	0x08	RW
Alarm1 Set Value Register Low	AL1SVL	0x09	RW
Alarm1 Set Value Register High	AL1SVH	0x0A	RW
Alarm1 Clear Value Register Low	AL1CVL	0x0B	RW
Alarm1 Clear Value Register High	AL1CVH	0x0C	RW
Temperature Configuration Register	TCR	0x10	RW
Temperature Data Register Low	TDRL	0x11	RO
Temperature Data Register High	TDRH	0x12	RO
Analog2 Configuration Register	AN2CR	0x14	RW
Low-Pass Filter2 Register	LPF2R	0x15	RW
Analog2 Data Register Low	AN2DRL	0x16	RO
Analog2 Data Register High	AN2DRH	0x17	RO
Alarm2 Configuration Register	AL2CR	0x18	RW
Alarm2 Set Value Register Low	AL2SVL	0x19	RW
Alarm2 Set Value Register High	AL2SVH	0x1A	RW
Alarm2 Clear Value Register Low	AL2CVL	0x1B	RW
Alarm2 Clear Value Register High	AL2CVH	0x1C	RW

Note: *Registers 0x02, 0x03, from 0x0D to 0x0F, 0x13, and from 0x1D to 0x1F are reserved registers which return 0x00 as a value.*

IV. General Register

ADDR – I2C Address Register

*Offset: **0x00***

Access: Read-Only

Bits	7	6	5	4	3	2	1	0
0x00	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
Default Value	1	0	0	0	0	0	1	0

- **Bits 7:0 – ADR7:0 – I2C Slave Address**
  These bits determine the unique address of the device on I2C bus. The default hardcoded address is 0x42.

FWVR – Firmware Version Register

*Offset: **0x01***

Access: Read-Only

Bits	7	6	5	4	3	2	1	0
0x01	FWV7	FWV6	FWV5	FWV4	FWV3	FWV2	FWV1	FWV0
Default Value	0	0	0	0	0	0	0	0

- **Bits 7:0 – FWV7:0 – Firmware Version**
  These bits determine the firmware version of the device. The value is hardcoded in the firmware and cannot be changed.

V. Analog Registers

AN1CR – Analog1 Configuration Register *Offset: **0x04***
AN2CR – Analog2 Configuration Register *Offset: **0x14***
Access: Read/Write

Bits	7	6	5	4	3	2	1	0
0x04, 0x14	-	-	-	-	-	-	LPEN	ANEN
Default Value	0	0	0	0	0	0	1	1

- **Bit 0 – ANEN – Analog Input Enable**
  When this bit is written to one, the analog input is enabled to sampling, then the register AN1DRH:L (AN2DRH:L) is updated. The sampling is done continuously when it is enabled. By writing it to zero, the register AN1DRH:L (AN2DRH:L) is frozen and hold their values. By default, the analog input is enabled to sampling.

- **Bit 1 – LPEN – Low-Pass Filter Enable**
  When this bit is written to one, the register ANDRH:L is averaged by a Low-Pass filter configured with the LPF1R (LPF2R) value. Writing it to zero disables the Low-Pass filter. By default, this bit is initialized with one.

LPF1R – Low-Pass Filter1 Register *Offset: **0x05***
LPF2R – Low-Pass Filter2 Register *Offset: **0x15***
Access: Read/Write

Bits	7	6	5	4	3	2	1	0
0x05, 0x15	-	-	-	-	LP3	LP2	LP1	LP0
Default Value	0	0	0	0	0	0	0	0

- **Bits 3:0 – LP3:0 – Low-Pass Parameter**
  These bits determine the parameter K of the low-pass filter. The LP max value is 15 and the min value is 0. However, the algorithm follows the formula K = LP + 1, then the parameter K is included between 1 and 16.
  Here is the Low-Pass filter algorithm implemented:

```
K = LP3:0 + 1;
LP_Filter = LP_Filter - (LP_Filter >> K) + ReadAnalog();
Result = LP_Filter >> K;
```

  The Low-Pass Filter was built from the recursive formula[1]:

$$output(t) = (1 - 2^{-K}) * output(t-1) + input(t)$$

  Bottom line: Higher is the parameter K, smoother is the proceeded analog value.

[1] Software implementation and theory are from A Simple Software Low-Pass Filter Suits Embedded-System Applications, by Barry Dorr.

AN1DRL, AN1DRH – Analog1 Data Registers
AN2DRL, AN2DRH – Analog2 Data Registers
Access: Read-Only

Offset: ***0x06, 0x07***
Offset: ***0x16, 0x17***

Bits	7	6	5	4	3	2	1	0
0x06, 0x16	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
Default Value	0	0	0	0	0	0	0	0

Bits	7	6	5	4	3	2	1	0
0x07, 0x17	-	-	-	-	-	-	AN9	AN8
Default Value	0	0	0	0	0	0	0	0

- **AN1DRL, AN2DRL – Bits 7:0 – AN7:0 – Analog Data Register Low**
  AN1DRH, AN2DRH – Bits 1:0 – AN9:8 – Analog Data Register High
  When sampling is complete, the result is found in these two registers. The analog value has a 10-bit resolution.
  The ANnDRH:L value can be the direct converted value from the Analog-to-Digital Converter OR the value from the output Low-Pass Filter. This configuration is set by LPEN (register ANnCR bit 1) and LP3:0 (register LPFnR).

VI. Alarm Registers

AL1CR – Alarm1 Configuration Register
AL2CR – Alarm2 Configuration Register
Access: Read/Write

*Offset: **0x08***
*Offset: **0x18***

Bits	7	6	5	4	3	2	1	0
0x08, 0x18	ALM2	ALM1	ALM0	ALME	-	-	ALO	ALEN
Default Value	0	0	0	0	0	0	0	0

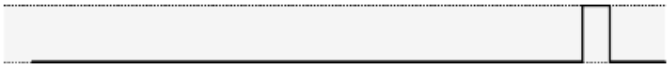
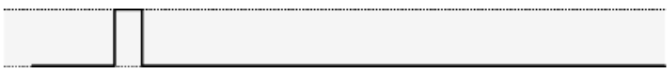
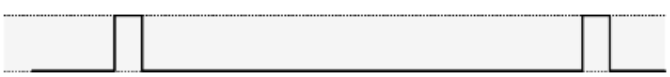
- **Bit 0 – ALEN – Alarm Enable**
  When this bit is written to one, Alarm1 (Alarm2) is enabled, and starts to check continuously the AL1SVH:L (AL2SVH:L) and AL1CVH:L (AL2CVH:L) registers. By writing it to zero, Alarm1 (Alarm2) is disabled, and ALO is cleared. By default, Alarm1 (Alarm2) is disabled.

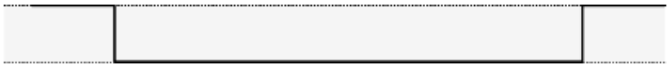
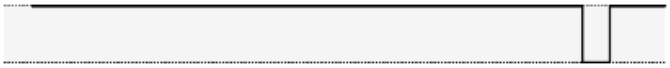
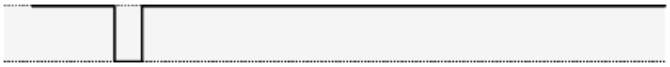
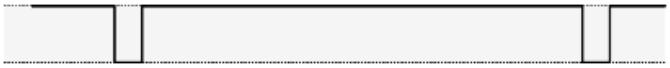
- **Bit 1 – ALO – Alarm Output**
  This bit has a Read-Only access. This is the output bit for Alarm1 (Alarm2) which is set by AL1SVH:L (AL2SVH:L) and cleared by AL1CVH:L (AL2CVH:L).

- **Bit 4 – ALME – Alarm Mode Enable**
  This bit enable the trigger for the hardware alarm output which is PB1 for Alarm1 and PB3 for Alarm2. This hardware output follows the waveform set by ALM2:0. When this bit is written to one, Alarm Mode is enabled, and ALM2:0 is used to set the mode selected. By writing it to zero, Alarm Mode is disabled, and ALM2:0 is ignored.

- **Bits 7:5 – ALM2:0 – Alarm Mode**
  These bits set the trigger mode for the hardware alarm output which is PB1 for Alarm1 and PB3 for Alarm2. The bits ALM1:0 determine if the output is a level (ALM1:0=0b00), a pulse such as falling edge (ALM1:0=0b01), raising edge (ALM1:0=0b10), dual edge (ALM1:0=0b11). The bit ALM2 determines the active low/high mode for both the pulse and level.

Output register value

ALM2:0	Hardware Output Signal	Comment
0 0 0		*Active high level*
0 0 1		*Pulse active high on falling edge*
0 1 0		*Pulse active high on raising edge*
0 1 1		*Pulse active high on raising and falling edge*

ALM2:0	Hardware Output Signal	Comment
1 0 0		*Active low level*
1 0 1		*Pulse active low on falling edge*
1 1 0		*Pulse active low on raising edge*
1 1 1		*Pulse active low on raising and falling edge*

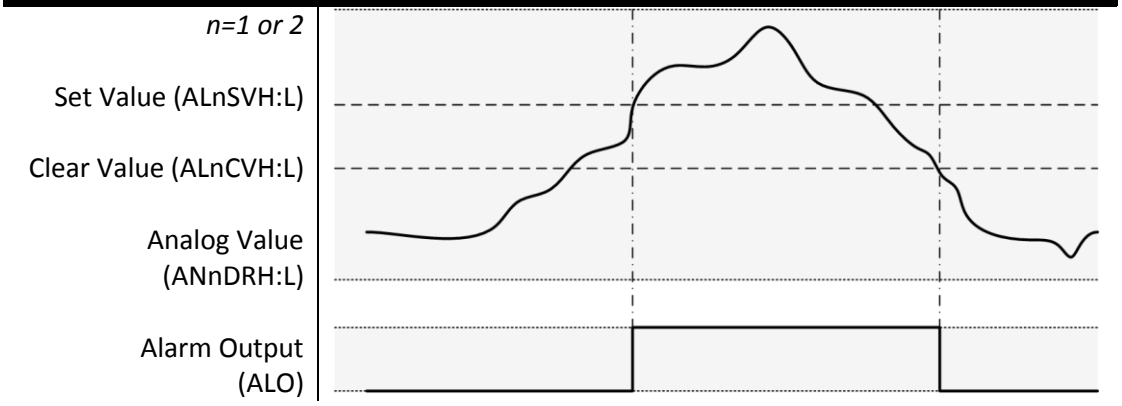
AL1SVL, AL1SVH – Alarm1 Set Value Registers *Offset: **0x09, 0x0A***
AL2SVL, AL2SVH – Alarm2 Set Value Registers *Offset: **0x19, 0x1A***
Access: Read/Write

Bits	7	6	5	4	3	2	1	0
0x09, 0x19	ALS7	ALS6	ALS5	ALS4	ALS3	ALS2	ALS1	ALS0
Default Value	0	0	0	0	0	0	0	0

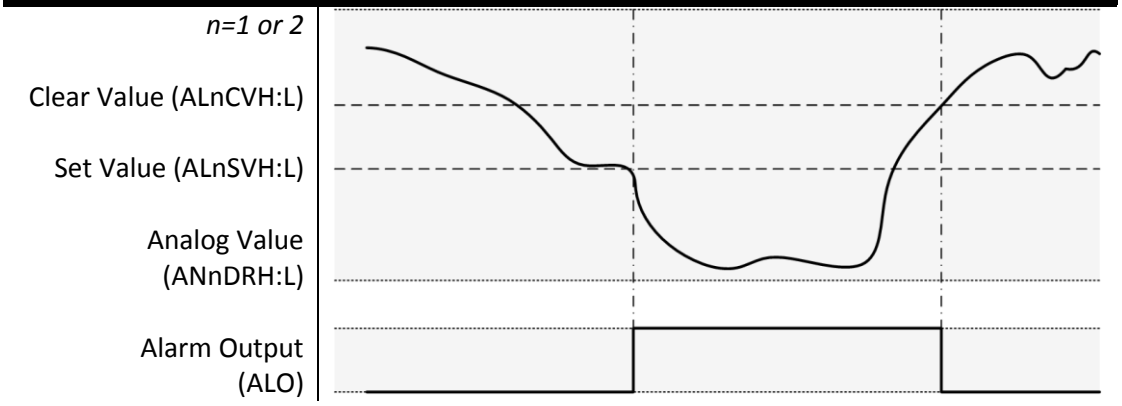
Bits	7	6	5	4	3	2	1	0
0x0A, 0x1A	-	-	-	-	-	-	ALS9	ALS8
Default Value	0	0	0	0	0	0	0	0

- **AL1SVL, AL2SVL – Bits 7:0 – ALS7:0 – Alarm Set Value Register Low**
  AL1SVH, AL2SVH – Bits 1:0 – ALS9:8 – Alarm Set Value Register High
  The Alarm1 (Alarm2) Set Value in AL1SVH:L (AL2SVH:L) is the trigger value for Alarm1 (Alarm2). When the analog value in AN1DRH:L (AN2DRH:L) is higher than the Set value, then Alarm1 (Alarm2) is triggered and ALO is high. ALO is cleared when AN1DRH:L (AN2DRH:L) is lower than the Clear value. The logic is identical if the Set value is lower than the Clear value.

Case 1 : Set Value > Clear Value



Case 2 : Clear Value > Set Value



AL1CVL, AL1CVH – Alarm1 Clear Value Registers
AL2CVL, AL2CVH – Alarm2 Clear Value Registers
Access: Read/Write

*Offset: **0x0B, 0x0C***
*Offset: **0x1B, 0x1C***

Bits	7	6	5	4	3	2	1	0
0x0B, 0x1B	ALC7	ALC6	ALC5	ALC4	ALC3	ALC2	ALC1	ALC0
Default Value	0	0	0	0	0	0	0	0

Bits	7	6	5	4	3	2	1	0
0x0C, 0x1C	-	-	-	-	-	-	ALC9	ALC8
Default Value	0	0	0	0	0	0	0	0

- **AL1CVL, AL2CVL – Bits 7:0 – ALC7:0 – Alarm1 Clear Value Register Low**
  AL1CVH, AL2CVH – Bits 1:0 – ALC9:8 – Alarm1 Clear Value Register High
  The Alarm1 (Alarm2) Clear Value in AL1CVH:L (AL2CVH:L) is the clear value for Alarm1 (Alarm2). When the analog value in AN1DRH:L (AN2DRH:L) is lower than the Clear value, then Alarm1 (Alarm2) is stopped and ALO is cleared. ALO is set when AN1DRH:L (AN2DRH:L) is higher than the Set value. The logic is identical if the Clear value is higher than the Set value.
  See example at AL1SVH:L (AL2SVH:L) registers.

VII. Temperature Registers

TCR – Temperature Configuration Register
Access: Read/Write

*Offset: **0x10***

Bits	7	6	5	4	3	2	1	0
0x10	TOS3	TOS2	TOS1	TOS0	TSNG	TUS1	TUS0	TEN
Default Value	0	0	0	0	0	0	0	1

- **Bit 0 – TEN – Internal Temperature Enable**
  When this bit is written to one, the internal temperature is enabled, and starts to sampling. By writing it to zero, the temperature sampling is disabled. By default, the internal temperature sensor is enabled.

- **Bit 2:1 – TUS1:0 – Temperature Unit Selection**
  These bits set the temperature unit for the internal temperature sensor. Available units are Celsius, Fahrenheit, and Kelvin. The raw value from the internal sensor is in Kelvin, but by default the temperature unit is set for Celsius conversion. The converted value is loaded in TDRH:L.

TUS1:0		Unit
0	**0**	*Celsius*
0	**1**	*Fahrenheit*
1	**X**	*Kelvin*

- **Bit 3 – TSNG – Offset Temperature Sign**
  When this bit is written to zero, the offset temperature has a positive sign. While a one means a negative sign.

- **Bit 7:4 – TOS4:0 – Offset Temperature Sensor**
  These bits determine the offset temperature for the internal temperature sensor. Valid values are from 0 to 15 Kelvin. The sign is set with the TSNG bit, thus the offset temperature range is ± 15 Kelvin. This offset will be applied on the raw temperature in Kelvin, just before the conversion in Celsius or Fahrenheit.

TDRL, TDRH – Temperature Data Registers

Offset: ***0x11, 0x12***

Access: Read-Only

Bits	7	6	5	4	3	2	1	0
0x11	TP7	TP6	TP5	TP4	TP3	TP2	TP1	TP0
Default Value	0	0	0	0	0	0	0	0

Bits	7	6	5	4	3	2	1	0
0x12	TP15	TP14	TP13	TP12	TP11	TP10	TP9	TP8
Default Value	0	0	0	0	0	0	0	0

- **TDRL – Bits 7:0 – TP7:0 – Temperature Data Register Low**
  TDRH – Bits 1:0 – TP9:8 – Temperature Data Register High
  When the temperature sampling and conversion are complete, the result is found in these two registers. The temperature unit is set by TCR.TUS1:0. The raw temperature value in Kelvin has a 10-bit resolution. These two registers are SIGNED.
  Here is the temperature sampling algorithm implemented:

```
RawTemperature = getSensorValue();
OffsetSign = TSNG ? -1 : 1; // get the offset sign
CorrectedTemperature = RawTemperature + OffsetSign x TOS4:0; //in kelvin
switch(TUS1:0):
      case 0:  ConvertedTemperature = ToCelsius(CorrectedTemperature)
      case 1:  ConvertedTemperature = ToFahrenheit(CorrectedTemperature)
      default: ConvertedTemperature = CorrectedTemperature
TDRH:L = ConvertedTemperature // signed value!!!
```

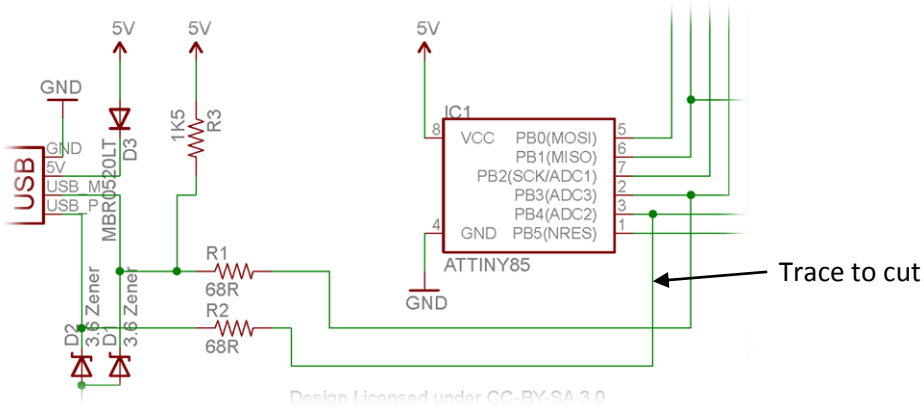
VIII. Hardware limitations

a. Analog Input 2 (AN2)

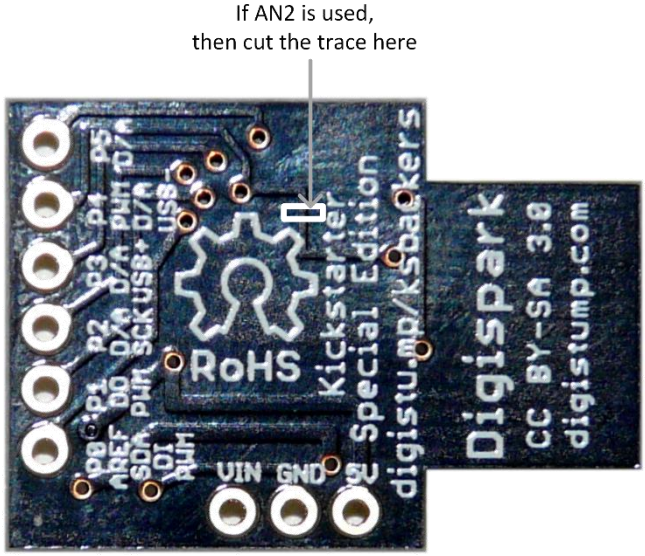
To use input AN2, the trace from USB_P has to be cut.

⚠ **This makes the USB port UNUSABLE!!!**
So program your Digispark before cutting the trace.

The Digispark schematic below shows that AN2 (P4) is connected to the USB_P pin.



The trace is located on the backside, above the HW open-source logo.



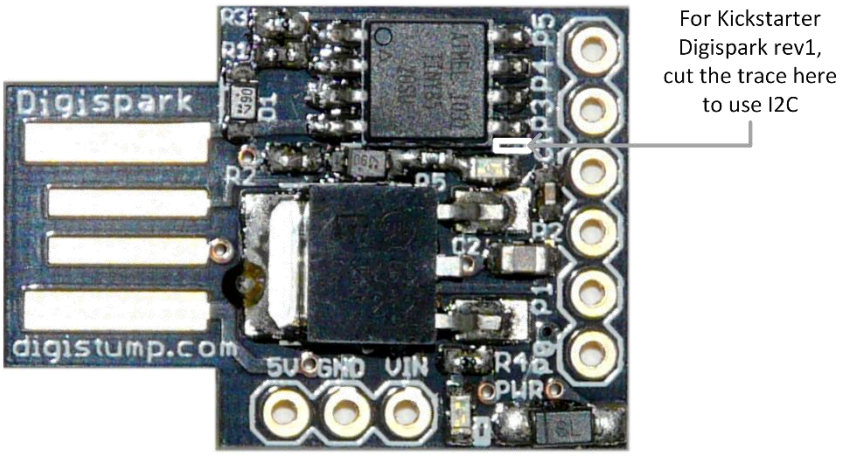
b. I2C on Kickstarter Digispark rev1

« You can identify your model by the presence of "rev2" on the top (the side with the gold connectors) of the USB end of the Digispark. » on [Digispark wiki]

With a rev1, the on-board LED is connected to P0. To use the SDA pin (P0) for I2C protocol, the trace from the LED has to be cut.

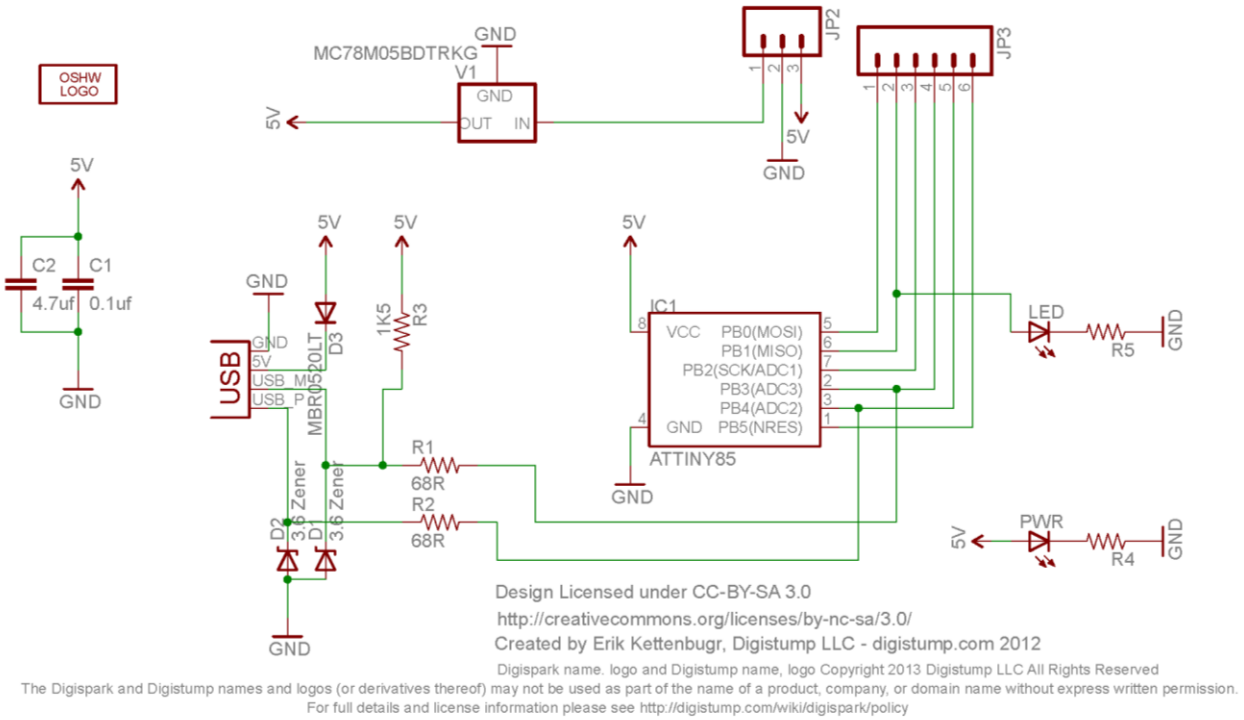
This is only valid for rev1. The other revisions have the on-board LED connected to P1. So for those latest versions, the LED can be used as an indicator light for AL1.

The photo below shows the trace to cut for rev1 board.



IX. Appendix

a. Digispark Schematic



Schematic version 2

b. Detailed Summary Register

Name	ID	Offset	Bit 7	6	5	4	3	2	1	0	Access
I2C Address Register	ADDR	0x00	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	RO
Firmware Version Register	FWVR	0x01	FWV7	FWV6	FWV5	FWV4	FWV3	FWV2	FWV1	FWV0	RO
Reserved Register	-	0x02	-	-	-	-	-	-	-	-	RO
Reserved Register	-	0x03	-	-	-	-	-	-	-	-	RO
Analog1 Registers											
Analog1 Configuration Register	AN1CR	0x04	-	-	-	-	-	-	LPEN	ANEN	RW
Low-Pass Filter1 Register	LPF1R	0x05	-	-	-	-	LP3	LP2	LP1	LP0	RW
Analog1 Data Register Low	AN1DRL	0x06	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	RO
Analog1 Data Register High	AN1DRH	0x07	-	-	-	-	-	-	AN9	AN8	RO
Alarm1 Registers											
Alarm1 Configuration Register	AL1CR	0x08	ALM2	ALM1	ALM0	ALME	-	-	ALO	ALEN	RW
Alarm1 Set Value Register Low	AL1SVL	0x09	ALS7	ALS6	ALS5	ALS4	ALS3	ALS2	ALS1	ALS0	RW
Alarm1 Set Value Register High	AL1SVH	0x0A	-	-	-	-	-	-	ALS9	ALS8	RW
Alarm1 Clear Value Register Low	AL1CVL	0x0B	ALC7	ALC6	ALC5	ALC4	ALC3	ALC2	ALC1	ALC0	RW
Alarm1 Clear Value Register High	AL1CVH	0x0C	-	-	-	-	-	-	ALC9	ALC8	RW
Reserved Register	-	0x0D	-	-	-	-	-	-	-	-	RO
Reserved Register	-	0x0E	-	-	-	-	-	-	-	-	RO
Reserved Register	-	0x0F	-	-	-	-	-	-	-	-	RO
Temperature Configuration Register	TCR	0x10	TOS3	TOS2	TOS1	TOS0	TSNG	TUS1	TUS0	TEN	RW
Temperature Data Register Low	TDRL	0x11	TP7	TP6	TP5	TP4	TP3	TP2	TP1	TP0	RO
Temperature Data Register High	TDRH	0x12	TP15	TP14	TP13	TP12	TP11	TP10	TP9	TP8	RO
Reserved Register	-	0x13	-	-	-	-	-	-	-	-	RO
Analog2 Registers											
Analog2 Configuration Register	AN2CR	0x14	-	-	-	-	-	-	LPEN	ANEN	RW
Low-Pass Filter2 Register	LPF2R	0x15	-	-	-	-	LP3	LP2	LP1	LP0	RW
Analog2 Data Register Low	AN2DRL	0x16	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	RO
Analog2 Data Register High	AN2DRH	0x17	-	-	-	-	-	-	AN9	AN8	RO
Alarm2 Registers											
Alarm2 Configuration Register	AL2CR	0x18	ALM2	ALM1	ALM0	ALME	-	-	ALO	ALEN	RW
Alarm2 Set Value Register Low	AL2SVL	0x19	ALS7	ALS6	ALS5	ALS4	ALS3	ALS2	ALS1	ALS0	RW
Alarm2 Set Value Register High	AL2SVH	0x1A	-	-	-	-	-	-	ALS9	ALS8	RW
Alarm2 Clear Value Register Low	AL2CVL	0x1B	ALC7	ALC6	ALC5	ALC4	ALC3	ALC2	ALC1	ALC0	RW
Alarm2 Clear Value Register High	AL2CVH	0x1C	-	-	-	-	-	-	ALC9	ALC8	RW
Reserved Register	-	0x1D	-	-	-	-	-	-	-	-	RO
Reserved Register	-	0x1E	-	-	-	-	-	-	-	-	RO
Reserved Register	-	0x1F	-	-	-	-	-	-	-	-	RO