Author: David Riedell
Date: April 20, 2016

Introduction

This app note will explain how to use GreenPAK to measure an analog voltage, convert it to a digital value using an 8-bit Analog to Digital Converter (ADC), transmit the data via I2C, and read the I2C command using an Arduino Uno Microcontroller. This 2 chip project uses the SLG46620 GreenPAK4 device and the SLG46531 GreenPAK5 device. The GreenPAK4 reads the Analog voltage, converts it to an 8-bit value, and outputs that value in parallel across 8 pins. The GreenPAK5 receives the 8 digital signals and transmits this information serially to a microcontroller via I2C.

Figure 1 shows the connections between the two GreenPAK devices, the bidirectional I2C SDA and SCL signals which are connected to the microcontroller, and the analog input voltage.

GreenPAK4 Design File

The GreenPAK4 chip has two input pins and 8 output pins. It receives an analog voltage at Pin 8 and a Power Down (PD) signal at Pin 10. Pins 12 through 19 transmit the 8 bits of ADC data to the GreenPAK5 chip, which is discussed in Section 3 of this App Note.

The analog signal at Pin8 is sent to the PGA (Programmable Gain Amplifier) whose settings are shown in Figure 3. It is important to note that the PGA must be manually set to "Power on," and the SPI (Serial to Parallel Interface) Parallel Output (Figure 6 6) must be manually enabled.

The PGA passes the analog signal over to the ADC, which transmits its data to the SPI over the orange 8-bit bus. The SPI block is clocked with the ADC's interrupt signal. Those 8 bits are then sent from the SPI Parallel Output Block to Pins 12 through 19 on the other side of the dual-matrix chip.

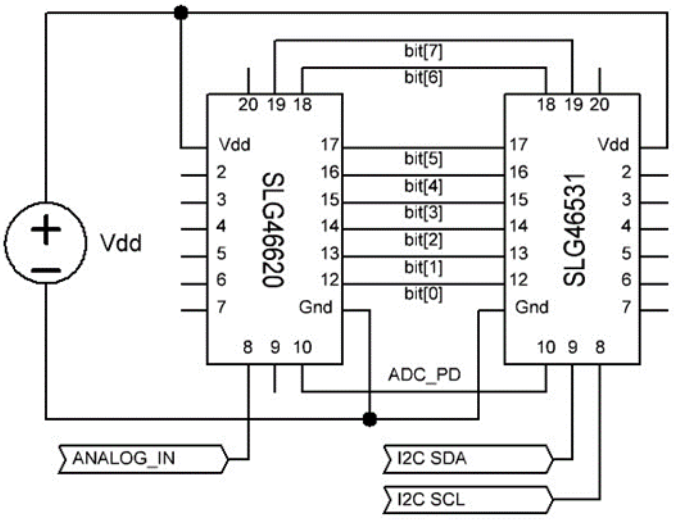

Figure 1. Circuit Diagram

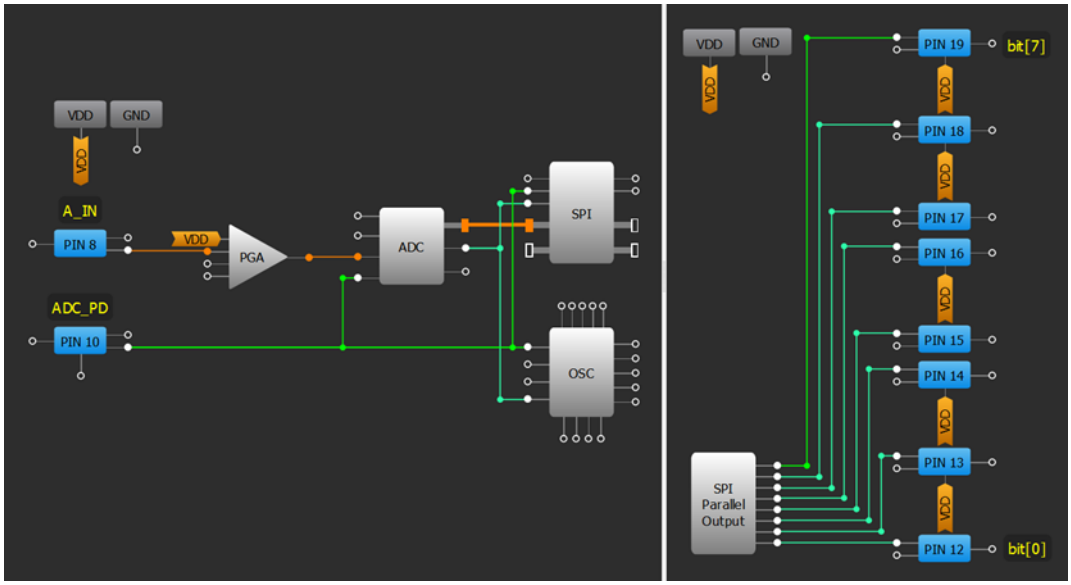

Figure 6. GreenPAK4 Design Block Diagram

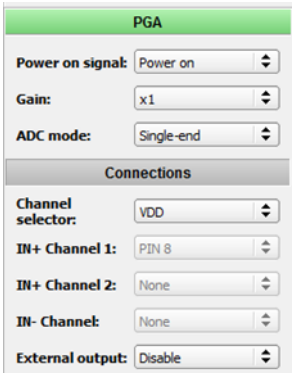

Figure 6. PGA Properties

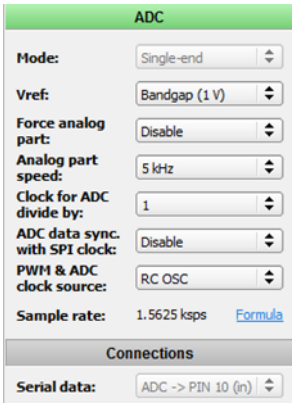

Figure 6. ADC Properties

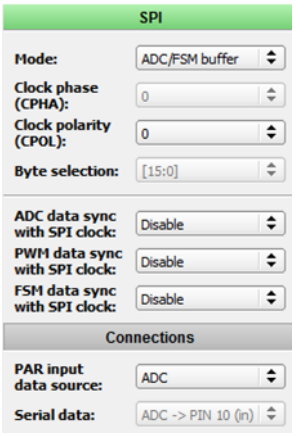

Figure 6. SPI Properties

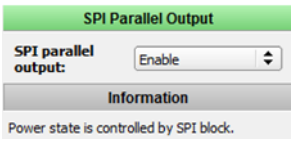

Figure 6. SPI Parallel Output Properties

GreenPAK5 Design File

The GreenPAK5 portion of this project has a very simple design. Pins 8 and 9 are dedicated I2C pins – Pin 8 carries the I2C clock (SCLK) and Pin 9 the I2C data (SDA). The first I2C Virtual Input signal is connected to digital output Pin 10, which allows us to power up or power down the ADC in the GreenPAK4 chip with a simple I2C command.

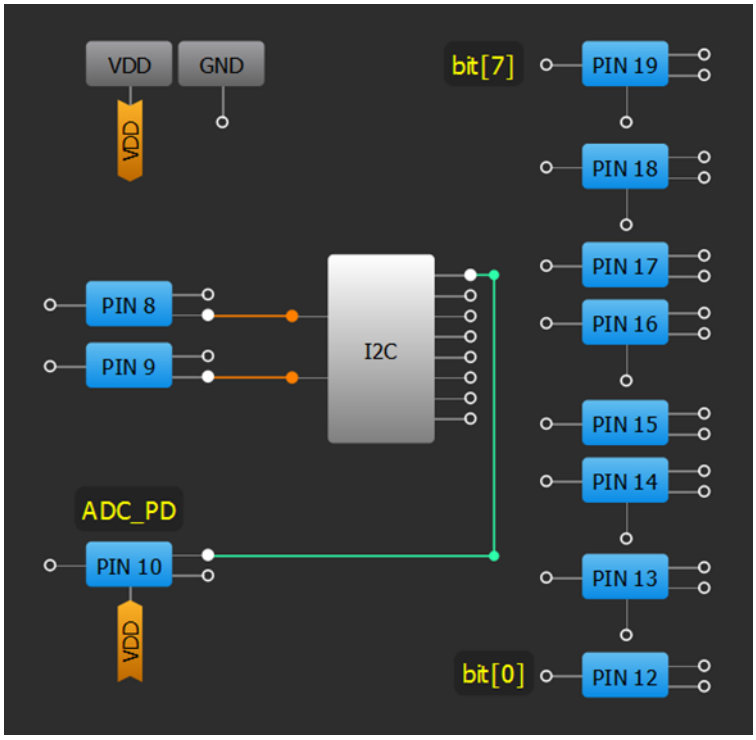

Figure 7. GreenPAK5 Design Block Diagram

Pins 12:19 are connected externally to the corresponding pin numbers on the GreenPAK4 device, as was illustrated in Figure 1. Each of the pins' input values can be read via I2C at address 0xF6.

Byte Address	Register Bit	Register Bit Definition
0xF6	Reg<1968>	Pin12 Digital Input
	Reg<1969>	Pin13 Digital Input
	Reg<1970>	Pin14 Digital Input
	Reg<1971>	Pin15 Digital Input
	Reg<1972>	Pin16 Digital Input
	Reg<1973>	Pin17 Digital Input
	Reg<1974>	Pin18 Digital Input
	Reg<1975>	Pin19 Digital Input

Table 1. Register Bit Definitions from p.156 of GreenPAK5 Datasheet

Arduino Uno Code

The code below was written for an Arduino Uno which functioned as our I2C master. This code enables the ADC, waits 600µs for valid ADC outputs, reads those ADC output values, and then disables the ADC to minimize power consumption. The Arduino Uno polls the ADC value once per second and prints the value via the Arduino Serial Monitor. For more information about interfacing with GreenPAK with I2C, please refer to **AN-1090**.

```
#include <Wire.h>

byte I2C_Byte_Addr = 0xF4;
byte ADC_Pins_Addr = 0xF6;

void setup() {
  Wire.begin();
  Serial.begin(9600);
  writeI2C(I2C_Byte_Addr, 0x01);
}

void loop() {
  writeI2C(I2C_Byte_Addr, 0x00);
  delayMicroseconds(600);
  uint8_t ADC_value =
readI2C(ADC_Pins_Addr, 0x00, 0x01);

  Serial.print("Value: ");
  Serial.println(ADC_value);
  writeI2C(I2C_Byte_Addr, 0x01);
  delay(1000);
}

void writeI2C(byte addr, byte data) {
  Wire.beginTransmission(0x00);
  Wire.write(addr);
  Wire.write(data);
  Wire.endTransmission();
}

uint8_t readI2C(byte addr, byte
slave_addr, byte bytes_to_read) {
  uint8_t value;
  Wire.beginTransmission(0x00);
  Wire.write(addr);
  Wire.endTransmission();
  Wire.requestFrom(slave_addr,
bytes_to_read);
  while(Wire.available()) { value =
Wire.read(); }
  return value;
}
```

Conclusion

In this App Note we described how to use a pair of GreenPAK devices to create an I2C-Readable Analog to Digital Converter. We used the provided code to interface between the GreenPAK5 and an Arduino Uno to enable the ADC and read the 8-bit digital value via I2C. The implementation of this simple design leaves many resources within the GreenPAK4 and GreenPAK5 chips available to perform other tasks.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.