SIGNAL CONDITION

Moon

Jellyfish

RASPBERRY PI:

Manta Ray

DISPLAYS:

Transparent Display

Moon

Jellyfish

POWER:

Battery

Shark

RESEARCH

LCD

Transferring Data

You are here: Home › Research › ATMEGA Core Temperature Sensor

ATMEGA Core Temperature Sensor

Abstract:

I recently stumbled across an interesting fact in the datasheet for the ATMEGA32u4, the microcontroller I am using for my Einstepper Project. I was surprised to find that Atmel had included a temperature sensor in the core of the device that you can read using the internal ADC. As it turns out, there are many megaAVR devices contain an internal temperature sensor. According to Atmel's product finder, these devices are:

- AT90PWM161
- AT90PWM81
- ATmega168A
- ATmega168P
- ATmega168PA
- ATmega16M1
- ATmega16U4
- ATmega328
- ATmega328P
- ATmega32M1
- ATmega32U4
- ATmega48A
- ATmega48P
- ATmega48PA
- ATmega64M1
- ATmega88A
- ATmega88P
- ATmega88PA

I was interested to see what temperature the microcontroller idled at

LOGIN

Username

Password

Remember Me

Login →

Register

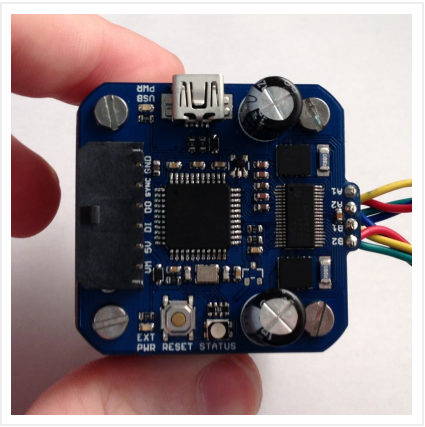
Lost Password

ATMEGA

Core Temp

and if this sensor could be used to detect the temperature of the room. By using a known temperature differential between the ambient air and core, it should be possible to calculate the ambient temperature. The Arduino Leonardo and Arduino Pro Micro use the ATMEGA32u4 so all the code below should work without modification. For other AVR boards that use a different chipset, minor modifications to the registers will be necessary.

Hardware:



I am using one of the boards from my Einstepper Project, but any board with one of the supported chips should work. There may be varying thermal characteristics between different packages (TQFP vs QFN) as well as different chips. These results are specifically for the ATMEGA32u4 in the TQFP package.

Software:

The first step was to setup the ADC to be able to read the internal temperature sensor. In order to use the temperature sensor correctly, you must set the ADC reference to the 2.56V internal reference, set the multiplexer to the temperature sensor, and enable the ADC. The code I used to configure the ATMEGA32u4's temperature sensor is as follows. This is specifically for the ATMEGA32u4 so if you are using a different chip, you should check the datasheet for the correct register values.

setupADC

```
1  void setupADC(){
2
3    //ADC Multiplexer Selection Register
4    ADMUX = 0;
5    ADMUX |= (1 << REFS1);  //Internal 2.56V Voltage Refe
6    ADMUX |= (1 << REFS0);  //Internal 2.56V Voltage Refe
7    ADMUX |= (0 << MUX4);  //Temperature Sensor - 100111
8    ADMUX |= (0 << MUX3);  //Temperature Sensor - 100111
9    ADMUX |= (1 << MUX2);  //Temperature Sensor - 100111
10   ADMUX |= (1 << MUX1);  //Temperature Sensor - 100111
11   ADMUX |= (1 << MUX0);  //Temperature Sensor - 100111
```

Here is the code I used to check the temperature sensor's value. It starts the ADC conversion, waits for it to finish and then returns the temperature in degrees Celsius. The value returned by the temperature sensor is in kelvin, so you need to subtract 273 from it to get the temperature in Celsius. The code below is not calibrated, which is something you need to do after you get the temperature sensor working.

```
1  int getTemp(){
2    ADCSRA |= (1 << ADSC);  //Start temperature conversion
3    while (bit_is_set(ADCSRA, ADSC));  //Wait for conversion
4    byte low  = ADCL;
5    byte high = ADCH;
6    int temperature = (high << 8) | low;  //Result is in kelv
7    return temperature - 273;
8  }
```

And here is the setup and loop I used:

```
1  void setup(){
2    //Start Serial Port
3    Serial.begin(9600);
4
5    setupADC();
6  }
7
8  void loop(){
9    Serial.print("Time: ");
10   Serial.print(millis());
11   Serial.print(" Core Temperature: ");
12   Serial.print(getTemp());
13   Serial.println(" C");
14
15   delay(1000);
16 }
```

This code should allow you to read the temperature sensor and print the temperature (in Celsius) to the serial port.

At this point, the temperature sensor has not been calibrated accurately, and according to the datasheet, the factory calibration is accurate to +- 10C. To calibrate the temperature, I held an ice cube against the ATMEGA32u4 (in a plastic bag so the board doesn't get wet) until the temperature reading stabilized around 7C. Because water has an enthalpy of fusion of 0C (by definition of Celcius), the ice should be exactly 0C. Assuming there is sufficient thermal conduction, the

core should be very close to 0C too. From this we know that the core temperature sensor reads 7C too high so I added an offset variable to the calculation as follows:

```
#define TEMP_OFFSET -7
int getTemp(){
  ADCSRA |= (1 << ADSC);  //Start temperature conversion
  while (bit_is_set(ADCSRA, ADSC));  //Wait for conversion
  byte low  = ADCL;
  byte high = ADCH;
  int temperature = (high << 8) | low;  //Result is in kelv
  return temperature - 273 + TEMP_OFFSET;
}
```

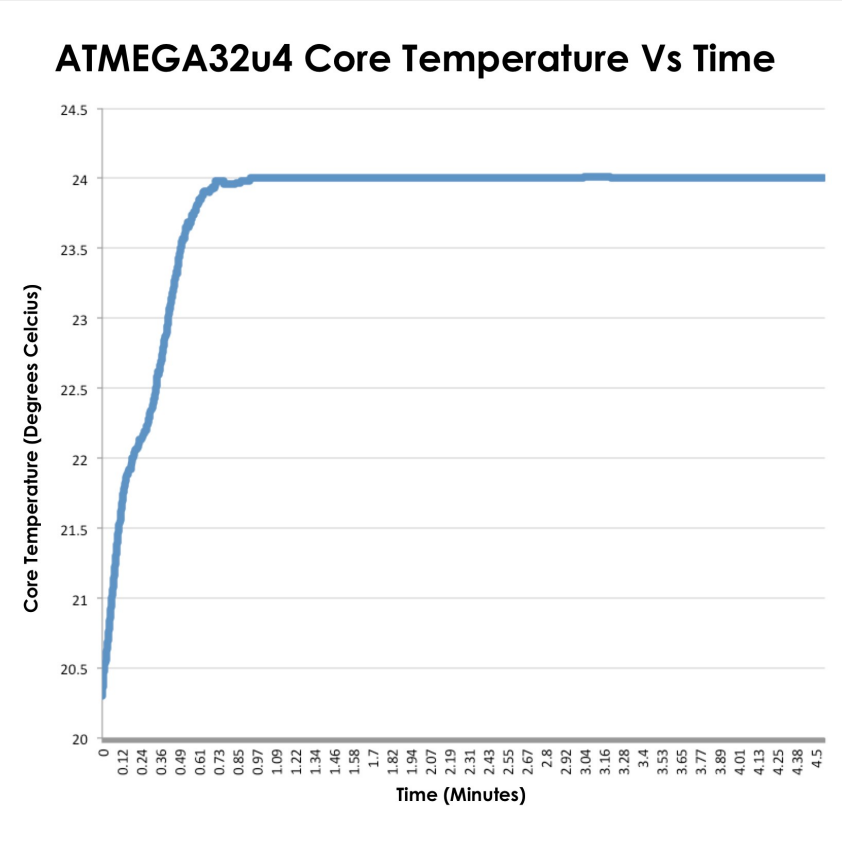
You will need to calculate this offset value for each chip since differences in manufacturing can result in the the chip reading up to 10C too hot or 10C too cold.

So here is the complete program (for ATMEGA32u4, Registers will be different for other chips):

```
#define TEMP_OFFSET -7

void setupADC(){
  cli();  //Disable global interrupts

  //ADC Multiplexer Selection Register
  ADMUX = 0;
  ADMUX |= (1 << REFS1);  //Internal 2.56V Voltage Refe
  ADMUX |= (1 << REFS0);  //Internal 2.56V Voltage Refe
  ADMUX |= (0 << MUX4);  //Temperature Sensor - 100111
  ADMUX |= (0 << MUX3);  //Temperature Sensor - 100111
  ADMUX |= (1 << MUX2);  //Temperature Sensor - 100111
  ADMUX |= (1 << MUX1);  //Temperature Sensor - 100111
  ADMUX |= (1 << MUX0);  //Temperature Sensor - 100111

  //ADC Control and Status Register A
  ADCSRA = 0;
  ADCSRA |= (1 << ADEN);  //Enable the ADC
  ADCSRA |= (1 << ADPS2);  //ADC Prescaler - 16 (16MHz

  //ADC Control and Status Register B
  ADCSRB = 0;
  ADCSRB |= (1 << MUX5);  //Temperature Sensor - 100111

  sei();  //Enable global interrupts
}

int getTemp(){
  ADCSRA |= (1 << ADSC);  //Start temperature conversio
  while (bit_is_set(ADCSRA, ADSC));  //Wait for convers
```

Results:

After logging the temperature for about 5 minutes and running the results through a 100 value running average filter, I generated the graph below.



One Response to "ATMEGA Core Temperature Sensor"

Tomek says:

September 26, 2014 at 9:28 pm

So you do know about the temp sensor! I was wondering why the einstepper had a temp sensor on the schematic :P

Log in to Reply

Leave a Reply

You must be logged in to post a comment.