Internal Temperature Sensor

New AVR chips (used in the Arduino) have an internal temperature sensor

AVR chips that have an internal temperature sensor:

- ATmega8 : No
- ATmega8L : No
- ATmega8A : No
- ATmega168 : No
- ATmega168A : Yes
- ATmega168P : Yes
- ATmega328 : Yes
- ATmega328P : Yes
- ATmega1280 (Arduino Mega) : No
- ATmega2560 (Arduino Mega 2560) : No
- ATmega32U4 (Arduino Leonardo) : Yes

The internal temperature is the temperature inside the chip, just like the cpu-temperature of a computer. If the Arduino is not sleeping, this temperature will increase. If output pins are used to drive current the internal temperature increases more.

ADC:

MUX3..0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8[1]
1001	(reserved)
1010	(reserved)
1011	(reserved)
1100	(reserved)
1101	(reserved)
1110	1.1V (V_{BG})
1111	0V (GND)

Note: 1. For temperature sensor.

ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Program:

```
 void setup()

{
  Serial.begin(9600);

  Serial.println(F("Internal Temperature Sensor"));
}

void loop()
{
  Serial.println(GetTemp());
  delay(1000);
}

double GetTemp(void)
{
  unsigned int wADC;
  double t;
  // Set the internal reference and mux.

  ADMUX = (_BV(REFS1) | _BV(REFS0)  | _BV(MUX3)); // Or 200;
  //((REFS1)  | (REFS0)  | (MUX3));
  //(_BV(REFS1) | _BV(REFS0)  | _BV(MUX3));
  //Serial.println((_BV(REFS1) | _BV(REFS0)  | _BV(MUX3)) );
  ADCSRA |= _BV(ADEN);  // enable the ADC

  delay(20);            // wait for voltages to become stable.


  ADCSRA |= _BV(ADSC);  // Start the ADC

  // Detect end-of-conversion
  while (bit_is_set(ADCSRA,ADSC));

  // Reading register "ADCW" takes care of how to read ADCL and ADCH.
  wADC = ADCW;
Serial.print("-->");
Serial.println(wADC);
```

```
  t = (wADC - 324.31 ) / 1.22;

  // The returned temperature is in degrees Celcius.
  return (t);
}
/* Bit 7 - ADEN: ADC Enable
 *  Bit 6 - ADSC: ADC Start Conversion
 *   Bit 5 - ADATE: ADC Auto Trigger Enabl
 * Bit 4 - ADIF: ADC Interrupt Flag
 * Bit 3 - ADIE: ADC Interrupt Enable
 * Bits 2:0 - ADPS[2:0]: ADC Prescaler Select Bits
 *
 *
 * ADCSRA – ADC Control and Status Register A
 *
 * ADSC-where the ADC is
  triggered manually by setting the ADSC bit to logic one in the ADCSRA register
 ADEN- enable the ADC


 ADMUX-set by bits REFS1 and REFS0 in the ADC Multiplexer
Select

 ADMUX-[REFS1] [REFS0] [ADLAR] [-] [MUX3] [MUX2] [MUX1] [MUX0]
 */
```