Features:

- Operating Voltage 3.3Volts.
- Two 10 bit SAR ADCs with 0V to 3.3V range.
- Two PWM Outputs for driving hobby RC Servo Motors, with 4000 steps of resolution. 0.6 millisecond pulse to 2.6 millisecond pulse. 2ms/4000 = .0005ms resolution
- SPI Interface, Standard 4-wire connection MOSI, MISO, S/ SCLK. Slave Mode Only.
- SPI maximum SCLK rate of 1 MHz.

General Description

The DAN777 IC is an expansion chip that adds two 10 bit ADC inputs and two PWM outputs for driving RC servo motors to your embedded system. It uses standard 4-wire SPI serial communication and the DAN777 is a SPI slave only device.

The two ADC channels are SAR (Successive Approximation Register) type ADCs with an input range of 0 to 3.3Volts. With 10 bits, that gives a input resolution of 3.3Volts/1024 steps. The DAN777 continuously samples the two ADC channels every 1 millisecond and the most current sample is transmitted to the SPI Master when the SPI Master issues a read/write command.

The two PWM outputs are specifically setup for driving RC servo motors. RC servo motors have an internal potentiometer for angle feedback. This allows the RC servo to move to a desired angle and hold there. To command an RC servo to a desired angle a repeating 3.3V pulse width drives the signal pin. This pulse needs to be repeated at least every 15ms. The DAN777 repeats the pulse every 20ms. Each brand of RC servo motor is slightly different when it comes to which pulse width commands a certain motor angle so a bit of experimenting is needed when driving a new RC servo motor. For example a pulse width around 1.6 ms should command the RC servo motor to its zero angle. A pulse width around 0.6 ms should command a -90 degree angle and a pulse width around 2.6 ms should command a 90 degree angle. Instead of thinking in terms of pulse width as the command, %duty cycle of a PWM signal can also be used to describe the input. The PWM signal has a carrier frequency of 50 Hz or 0.02s period and a range of duty cycles from 3% to 13%

Device Pins

Vcc: 3.3V power for the IC
Gnd: Ground of the 3.3V power
ADCIN1: ADC channel one’s input pin. Accepts a voltage in the range of 0 to 3.3 volts. Value communicated over SPI is an integer with the range of 0 to 1023 where 0 equals 0 volts and 1023 equals 3.3volts.
ADCIN2: ADC channel two’s input pin. Accepts a voltage in the range of 0 to 3.3 volts. Value communicated over SPI is an integer with the range of 0 to 1023 where 0 equals 0 volts and 1023 equals 3.3volts.
**RC1**: RC servo motor command output one. RC servos are driven by pulse widths in the range of 0.6 ms to 2.6 ms. These pulses are repeated every 20 ms. So you can also think of this as a PWM signal with carrier frequency 50Hz and duty cycles ranging from 3% to 13%. The value communicated over SPI is an integer with the range of 1200 to 5200 where 1200 equals 3% and 5200 equals 13% duty cycle.

**RC2**: RC servo motor command output two. RC servos are driven by pulse widths in the range of 0.6 ms to 2.6 ms. These pulses are repeated every 20 ms. So you can also think of this as a PWM signal with carrier frequency 50Hz and duty cycles ranging from 3% to 13%. The value communicated over SPI is an integer with the range of 1200 to 5200 where 1200 equals 3% and 5200 equals 13% duty cycle.

**SS**: Slave Select, Active Low. Slave select must be held for the entire SPI communication to the DAN777. If high the DAN777 is not selected and no communication can occur.

**SCLK**: SPI CLK Clock is normally low when no communication is in progress. During communication, both the SPI master and SPI slave read in a new bit on the rising edge of the SCLK. The SPI master reads the new bit (1 or 0) of the MISO pin. The SPI slave reads the new bit (1 or 0) of the MOSI pin.

**MOSI**: SPI Slave Input pin. RCservo commands are received on this pin.

**MISO**: SPI Slave Output pin. RCservo commands are transmitted on this pin.

**Live Stat**: Is the DAN777 running status. This is a useful pin that indicates if the DAN777 is operating. Simply wire an LED in series with a current limiting resistor to this pin. When powered the DAN777 will blink on and off this LED.

**Reset**: Used to reset the program running on the DAN777 back to the beginning of its code. Wire this pin to a switch and pullup resistor. This is a nice feature when debugging the SPI Master’s code. Whenever the SPI Master code is restarted the DAN777 should be reset by pressing this button, in order that the SPI communication is in sync.

**Rsrv**: Reserved for future use in new releases of the DAN777.

### Typical Wiring for DAN777

![Typical Wiring for DAN777](image_url)
Registers in the DAN777

ADC1: Value between 0 and 1023 so need two bytes to communicate its value
   ADC1MSB: (8 bit value)
   ADC1LSB: (8 bit value)
   ADC1 voltage reading can be found by combining these two 8 bit values, received over SPI, into one 16 bit value. This combined 16 bit value, has a range from 0 to 1023 where 0 is 0V and 1023 is 3.3V.
   \[ \text{ADC1 value} = (\text{ADC1MSB value} << 8) | \text{ADC1LSB} \]

ADC2: Value between 0 and 1023 so need two bytes to communicate its value
   ADC2MSB: (8 bit value)
   ADC2LSB: (8 bit value)
   ADC2 voltage reading can be found by combining these two 8 bit values, received over SPI, into one 16 bit value. This combined 16 bit value, has a range from 0 to 1023 where 0 is 0V and 1023 is 3.3V.
   \[ \text{ADC2 value} = (\text{ADC2MSB value} << 8) | \text{ADC2LSB} \]

RC1: Value between 1200 and 5200 so need two bytes to communicate its value
   RC1MSB: (8 bit value)
   RC1LSB: (8 bit value)
   In terms of a 50Hz carrier frequency PWM signal, 1200 = 3% duty cycle and 5200 = 13% duty cycle. 3% duty cycle approximately commands an angle of -90 degrees. 8% approximately commands an angle of 0 degrees. 13% approximately commands an angle of 90 degrees. To divide the PWM command into its MSB and LSB:
   \[ \text{RC1msb} = (\text{RC1 value} >> 8) \& 0xFF \]
   \[ \text{RC1lsb} = \text{RC1 value} \& 0xFF; \]
   If the DAN777 receives a value less than 1200, the value of 1200 or 3% duty cycle is commanded. If the DAN777 receives a value greater than 5200, the value of 5200 or 13% duty cycle is commanded.

RC2: Value between 1200 and 5200 so needs two bytes to communicate its value
   RC2MSB: (8 bit value)
   RC2LSB: (8 bit value)
   In terms of a 50Hz carrier frequency PWM signal, 1200 = 3% duty cycle and 5200 = 13% duty cycle. 3% duty cycle approximately commands an angle of -90 degrees. 8% approximately commands an angle of 0 degrees. 13% approximately commands an angle of 90 degrees. To divide the PWM command into its MSB and LSB:
   \[ \text{RC2msb} = (\text{RC2 value} >> 8) \& 0xFF \]
   \[ \text{RC2lsb} = \text{RC2 value} \& 0xFF; \]
   If the DAN777 receives a value less than 1200, the value of 1200 or 3% duty cycle is commanded. If the DAN777 receives a value greater than 5200, the value of 5200 or 13% duty cycle is commanded.
SPI Interface

Each time the SPI master needs to communicate with the DAN777 the above timing diagram must be followed:

1. SS\ must be pulled low by the SPI Master.
2. 0xDA must be sent as the first 8 bit value. During this transmission of 0xDA, the DAN777 sends nothing important back to the Master so this byte can be discarded once read on the Master's end.
3. SPI master sends the most significant byte of the RCservo1 command, RC1MSB. During this transmission, ADC1MSB is sent to the SPI Master.
4. SPI master sends the least significant byte of the RCservo1 command, RC1LSB. During this transmission, ADC1LSB is sent to the SPI Master.
5. SPI master sends the most significant byte of the RCservo2 command, RC2MSB. During this transmission, ADC2MSB is sent to the SPI Master.
6. SPI master sends the least significant byte of the RCservo2 command, RC2LSB. During this transmission, ADC2LSB is sent to the SPI Master.
7. SS\ must be pulled high by the SPI Master.