LABORATOIRE DE MICRO-INFORMATIQUE EPFL and DIDEL

File www.didel.com/doc/DocSharp.pdef

0.97

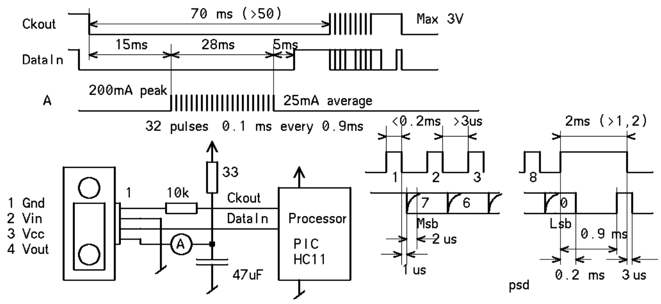
The Sharp GP2D02 Distance Sensor

The Sharp GP2D02 is excellent for measuring distances between 10cm and 80cm, but it has several drawbacks:

- Below 8 to 10 cm, the given value are erratic and correspond to high distances
- Above 1m, or in front of a mirror, the device gives erratic values
- The device is terribly slow with a 50 to 70 ms cycle time. This means that you will not measure distances so frequently. Your 2m/s robot will get a new distance every 20 cm only.
- The documentation is incredibly unprecise. You can get the specs from http://www.sharp.co.jp/ecg/sys/gp2d02/gp2d02-fea.html but the functional timing diagrams are imprecise, no min/max timing are given, electrical parameters are incomplete, power-off is unclear.
- Power consumption is 25 mA average, but one must take care of the 200 mA peaks when the distance is measured.
- The clock input Vin is not compatible with processor outputs. Electrically, it something never encountered on other devices. There is no electrical model, and only an unclear mention of a 0.3 mA max currant for Vin=0. A 10 kOhm resistor must be used to limit the current. A diode with the anode on the GP2D02 side do the same function.
- The old model without the fixation holes does not have the same timings.

The GP2D12 has an analogue signal out. It is slightly faster, and has been well characterized by J.C. Zufferey: see dmtwww.epfl.ch/~jzuffere/SharpGP2D12_E.html

For the timing of the GP2D02, we document in next figure what we have measured on one device.



The device starts operating with a quite long 70 ms (specified) zero level on the Vin clock input. That pulse is long because the circuit needs 15 ms to power-up, then send 32 pulses to its internal LED and do probably some filtering on this 1 kHz signal, and finally prepare the measure for about 5 ms; Vo/DataIn is activated to indicate the circuit is ready. That not documented feature saves 15 to 20 ms. The infrared pulses were made visible with a 4 Ohm resistor on the +5V. The current increases to 200 mA when the internal LED is activated, creating a 1V voltage drop visible at the scope. If you do not want to have this current surge coming back to your power supply, you have to store locally 0.1 ms x 200 mA = 2 x 10-5 Coulombs and regenerate it in 0.8 ms. If you accept a voltage drop of 0.5 V (minimum voltage is documented as 4.4V), the capacitance is the number of Coulombs divided by 0.5, that is about 40 μF. If you do this, you can insert a serial resistor of about 40 Ohm and have a continuous voltage drop of 0.2V (average current is 25 mA plus recharging the capacitor 25 mA). If you do not have the large capacitor next to the Sharp, you had better have good ground lines, and avoid to power several Sharp synchronously.

After the 50 to 70 ms of the start pulse, you can provide clocks and get the data. Clock high can be apparently as low as 1 μs, but we recommend 2 μs minimum. Data is provided after the negative clock edge, 1 μs later for a 1 to zero transition, 3 to 4 μs later for a low to high transition (12 kOhm internal pull-up). Data is always one during the first clock. The first falling clock edge makes the most significant bit (Msb) appear. It is necessary to wait 4 μs before sampling it.

After you have got the 8th data bit, the Sharp needs a long 2 ms clock pulse for undiscovered reasons. The last data bit (the Lsb) desappear after 0.2 ms and the data line Vo rises again after 0.9 ms, when the device switch in low power mode, which is only several microAmps. Specification requires a 2 ms pulse. 1.2 ms was measured for our test circuit. Again Vo/DataOut can be used as a handshake signal to continue. It returns to zero when the Vo/clock signal returns to zero to start a new cycle. If you increase the duration of that "stop" positive pulse, you will decrease the average power consumption.

Programming

If you have no other task to do in parallel, e.g. for checking the operation as we did to understand the timings and measure the response time not given by the manufacturer, delay loops are good. The structure of the program with delay is

```
Repeat for ever
    CkOff deactivate Vin/Clock
    Wait 70 ms (worst case specs)
    Repeat 8 times
        Clock pulse 2 µs positive pulse on Vin/Clock
        Wait 3 µs minimum
        Copy the bit read on Vo/DataIn to the carry
        Rotate Left through the Carry the variable Data
    End Repeat
    ClkOn activate the clock
    Wait 2 ms or more if there is no need to do frequent measures
End Repeat
```

If you use the handshake mode, not documented by Sharp, wut which clearly appear on their timing diagrams, you can save 20ms (see PSDtest2.asm program). The given timings have been measured with an oscilloscope on the device we tested, plus some margin.

```
Repeat for ever
    CkOff disactivate Vin/Clock
    Wait 0.1ms
    Wait until Vo/DataIn goes high
    Repeat 8 times
        Clock pulse 2 µs positive pulse on Vin/Clock
        Wait 3 µs minimum
        Copy the bit read on Vo/DataIn to the carry
        Rotate Left through the Carry the variable Data
    End Repeat
    ClkOn activate the clock
    Wait 0.2 ms
    Wait until Vo/DataIn goes high
End Repeat
```

Synchronous programming

A simple 8-pin PIC is adequate for reading the Sharp and transfering the value over a serial line, using Serial, I2C, SPI, Frequency, PWM, or analog output (see "Serial transfers for sensors").

Serial transfers are possible up to 9600 b/s. I2C is not feasible if the I2C is at its nominal speed of 100 kb/s. On a HC11 not having I2C special hardware, it is possible to do transfers at 10 kb/s. SPI is possible also at 10 kb/s. Frequency is quite easy: the frequency can e.g. be programmed between 100 Hz (for 80 cm) and 1 kHz (for 8 cm) using a linear scale with 10% worst case precision, or between 100 Hz and 10 kHz with a 1% precision on a logarithmic scale.

The PWM output is documented below. Frequency is only 200 Hz, with 64 PWM values. Distance can be measured by the host processor every 5 ms, with 2% precision, using a timer able to measure pulses between 0.07 ms and 5 ms. The PWM value can be filtered by a RC network (10 kOhm resistor, 1 µF capacitor) and measured by some analog input of the microprocessor. RC value must be carefully checked on the scope: too large, response time will be increased. Too low, ripples will increase the noise.

Synchronous task for the PSD are the following. Cycle time can be as low as 50 microsecond.

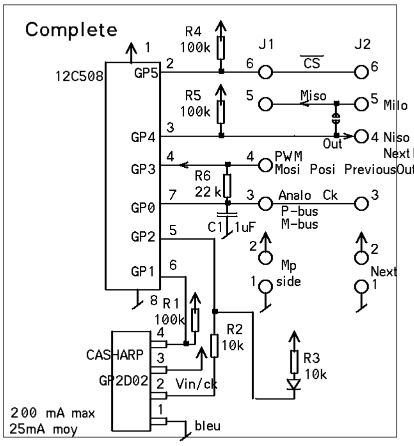
PsdTask0	ClkOff
PsdTask1	Wait for DataIn high, prepare :8 counter
PsdTask2	Clkpulse and get data, prepare a 0.2ms wait
PsdTask3	ClkOn, wait
PsdTask4	Wait for DataIn high, return to Task0

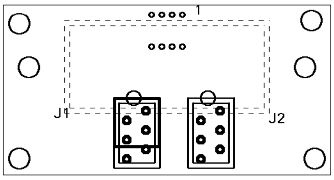
Data given by the PSD (between 230 and 70) is shifted right twice to reduce the size of the linearisation table to 64 entries (there is space on the PIC for the complete table if correctly placed in ROM, but we do not need such a precision). The table define the PWM value, which is passed to the PWM routine, computed at every cycle.

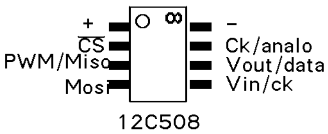
LAMI PSD module

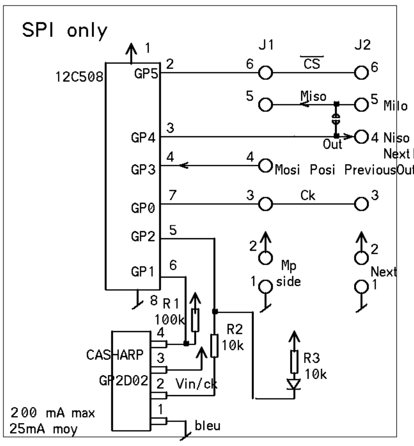
The LAMI "psdspi" PC board carries a cheap PIC 12C508 which can be programmed for several modes of operations:

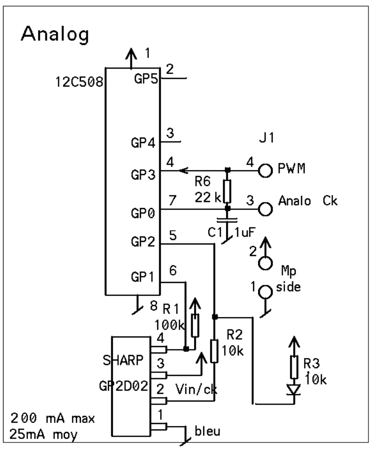
- PWM output and analog output proportional to the distance
- SPI output (8-bit slave), with the constraint that the SPI clock is maximum 50 kb/s, but with the advantage that several modules can be cascaded.
- Petra 1-wire bus (or Maxim or Dallas)

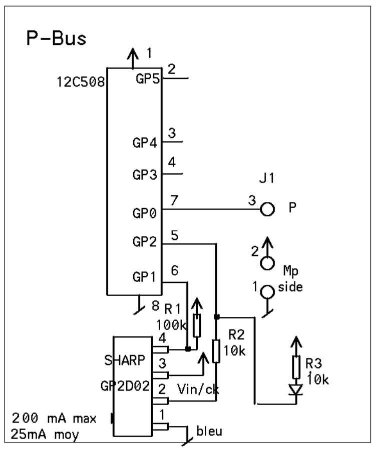


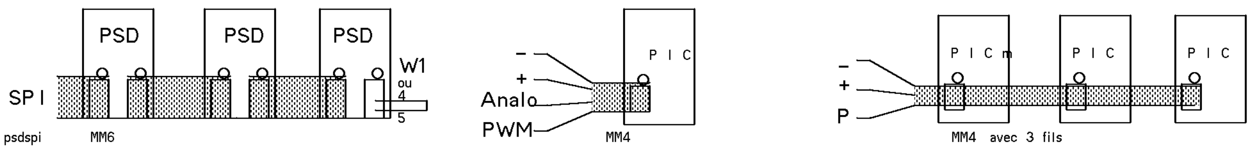












The ISR PSR module

Developed by Roger Meier and Unai Viscarret, this 43x27mm module has an I2C and RS232 interface and is available at EPFL-DMT-ISR-LSL (Ralph Piguet or Michel Lauria).