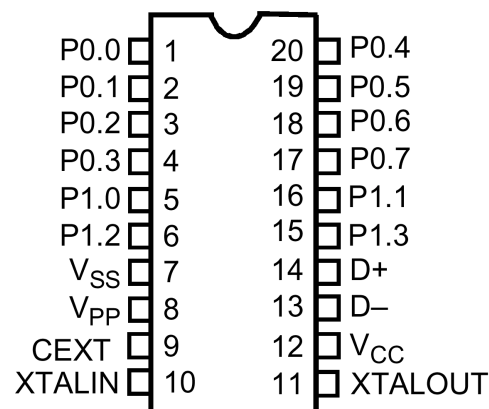


# I/O Solution

- 12 I/O lijnen, elk bruikbaar als ingang of als uitgang.
- Programmeerbare pull-up weerstand op elke lijn, per lijn apart te kiezen.
- In 16 stappen programmeerbare uitgangs stroom, per lijn individueel te programmeren, voor het direct aansluiten van LED's etc.
- Vormt met slechts twee externe componenten een complete I/O bouwsteen voor op de USB bus.
- Software geschikt voor Windows 98SE, ME, 2000 en XP.



1	Poort 0 bit 0	Poort 0 bit 4	20
2	Poort 0 bit 1	Poort 0 bit 5	19
3	Poort 0 bit 2	Poort 0 bit 6	18
4	Poort 0 bit 3	Poort 0 bit 7	17
5	Poort 1 bit 0	Poort 1 bit 1	16
6	Poort 1 bit 2	Poort 1 bit 3	15
7	Vss (0 Volt)	D+ (USB bus)	14
8	Vpp (verbind deze met 0V)	D- (USB bus)	13
9	Cext (open-drain suspend uitgang)	Vcc (+5 Volt)	12
10	XTALIN aansluiten op resonator	XTALOUT aansluiten op resonator	11

## Inleiding

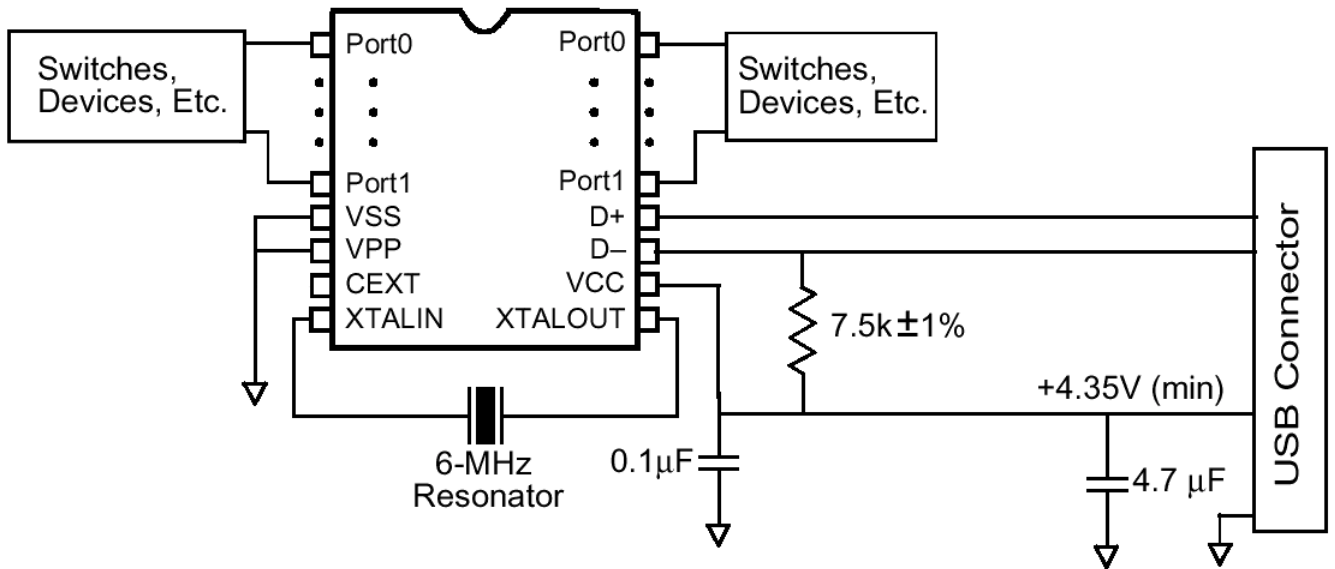
De I/O Solution is gebaseerd op een CY7C63001A microcontroller van Cypress ([www.cypress.com](http://www.cypress.com)) Deze microcontroller heeft een speciaal programma gekregen om zijn werk als I/O Solution te kunnen doen. Het programma voor de microcontroller kan niet gewijzigd worden.

Bij aansluiten aan een PC zorgt het ingebouwde programma van de I/O Solution er voor, dat deze door de PC automatisch wordt herkend. De I/O Solution heeft geen specifieke drivers nodig, de benodigde drivers maken standaard deel uit van Windows 98 (SE), ME, 2000 en XP. Windows 95 ondersteunt de I/O Solution niet.

De I/O Solution kan vanuit de PC eenvoudig worden aangestuurd.

## Aansluiten

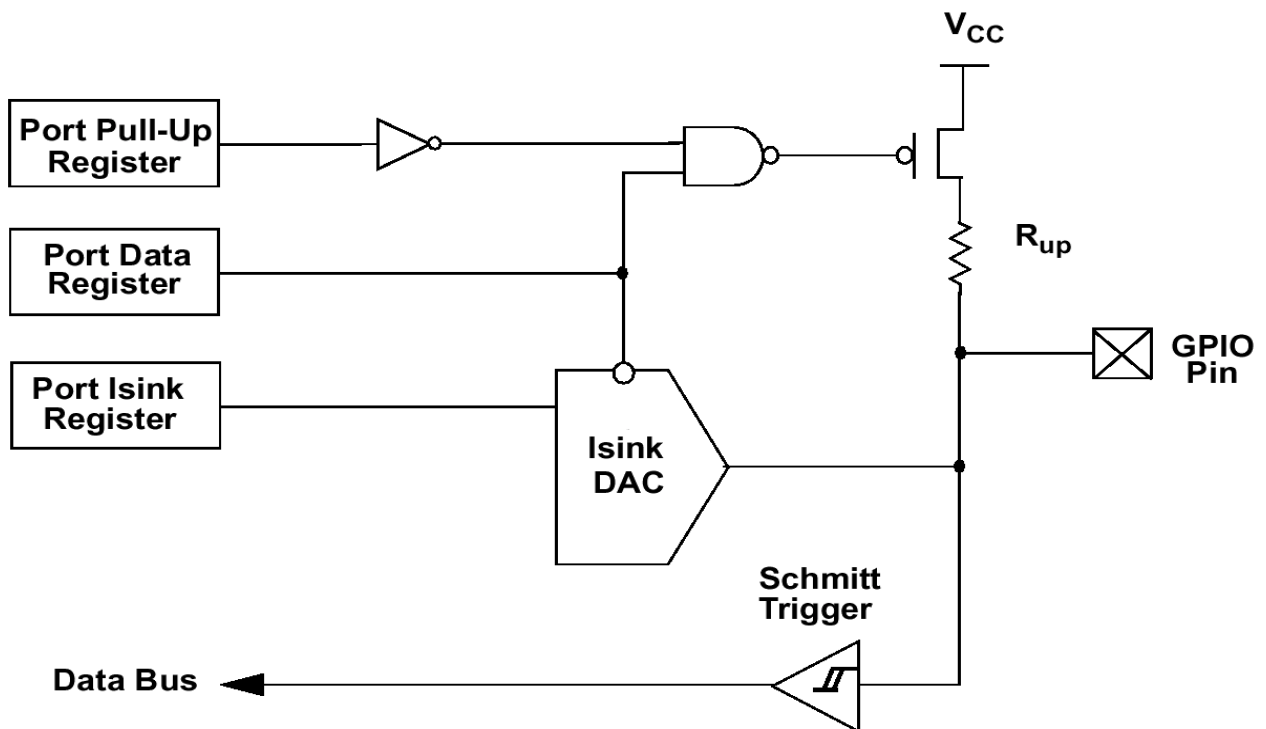
Om de I/O Solution op de USB bus aan te sluiten is niet meer nodig dan een +5 Volt voeding, die van de USB bus kan worden afgenomen, en de twee USB datalijnen. Een weerstand van 7.5 kΩ tussen +5 Volt en de D- lijn van de USB bus, vertelt de USB bus dat er een low-speed apparaat is aangesloten. Een keramische resonator van 6MHz is genoeg om de I/O Solution te laten werken. Een paar ontkoppel condensatoren kunnen worden toegevoegd om het geheel storingsvrij te laten functioneren.



## I/O poorten

Er zijn twee poorten van elk 8 bits. Van poort 0 zijn alle 8 de lijnen naar buiten uitgevoerd. Van poort 1 zijn 4 lijnen niet van aansluitpinnen voorzien. In totaal heeft de I/O Solution dus 12 lijnen die elk als ingang of uitgang gebruikt kunnen worden.

Het interne schema van een poort ziet er als volgt uit:



Hierin zijn de volgende functies te herkennen:

1. Een schmitt-trigger ingangspoort die wordt gebruikt om het werkelijke logische niveau van de I/O pin te meten.
2. Met een pull-up weerstand, die naar believen is te activeren, kan de I/O pin intern een gedefinieerd 'hoog' niveau gegeven worden.
3. Met een DAC kan de stroom worden vastgelegd die de I/O poort in vloeit, als deze via het data register 'laag' gemaakt wordt.

Pull-up register	Data register	
0	0	De pin is laag, de stroom die de pin in loopt richting 0V is vastgelegd door de instelling van de DAC (16 standen) <sup>-</sup>
0	1	De pin wordt intern naar +5 V getrokken door de inwendige pull-up weerstand.
1	0	De pin is laag, de stroom die de pin in loopt richting 0V is vastgelegd door de instelling van de DAC (16 standen) <sup>-</sup>
1	1	De pin is een hoog-ohmige ingang.

\* Poort 0 accepteert ingangs stromen tot nominaal 1mA, poort 1 accepteert ingangs stromen tot nominaal 20mA. De stroom voor 4 pinnen van poort 0 (P0.7, P0.6, P0.5 en P0.4) staat vast ingesteld op de maximale waarde. De stroom voor de andere 4 pinnen, en voor de 4 pinnen van poort 1, kan via de PC voor elke pin apart geprogrammeerd worden.

Voorbeelden:

- Een LED kan rechtstreeks worden aangesloten tussen een I/O pin van poort 1 en de +5 Volt. De stroom die door de LED moet gaan lopen kan in 16 standen worden ingesteld. Dit is niet alleen handig voor displays ed, maar ook voor de LED in een opto-coupler. Met een opto-coupler kan op veilige wijze bijvoorbeeld een relais worden geschakeld.

Pull-up register	Data register	Uitgangs stroom
1	0: LED aan, 1: LED uit	Instellen op LED stroom

- Een drukknop of een open-collector uitgang van bijvoorbeeld een opto-coupler kan zonder meer worden aangesloten op elk van de pinnen. De interne pull-up weerstand zorgt voor een gedefinieerd logisch niveau als de ingang niet geactiveerd is.

Pull-up register	Data register	Uitgangs stroom
0	1	nvt

- Een ingang van een (HC)TTL-compatibel logisch IC kan rechtstreeks worden aangestuurd door elk van de pinnen.

Pull-up register	Data register	Uitgangs stroom
0	1= logisch '1', 0 = logisch '0'	maximaal

- Een uitgang van een (HC)TTL-compatibel logisch IC kan rechtstreeks worden ingelezen door elk van de pinnen

Pull-up register	Data register	Uitgangs stroom
1	1	nvt

- Een NPN darlington transistor kan direct worden aangesloten op een pin naar keuze. Met de emitter van de transistor aan de 0 V, zorgt de interne pull-up weerstand voor basis stroom. Met de transistor kan een zwaardere belasting geschakeld worden. Deze belasting wordt aangebracht tussen een externe voeding en de collector van de transistor.

Pull-up register	Data register	Uitgangs stroom
0	1: Transistor geleidt, 0: Transistor spert	maximaal

## Specificaties I/O poorten

Parameter	Min	Max	Eenheid	Opmerkingen
Pull-up weerstand	8	25	kΩ	
Minimum stroom (instelling 0) Poort 0	0.1	0.3	mA	Vuit = 2.0 Volt
Maximum stroom (instelling 15) Poort 0	0.5	1.5	mA	Vuit = 2.0 Volt
Minimum stroom (instelling 0) Poort 1	1.6	4.8	mA	Vuit = 2.0 Volt
Maximum stroom (instelling 15) Poort 1	8	24	mA	Vuit = 2.0 Volt

Maximum stroom (instelling 15) Poort 1	5		mA	Vuit = 0.4 Volt
Maximum stroom voor alle pinnen van Poort 0		10	mA	Alle stromen gesommeerd
Maximum stroom voor alle pinnen van Poort 1		60	mA	Alle stromen gesommeerd
Threshold spanning Schmitt-triggers	45%	65%	Vcc	2.25 ... 3.25V bij Vcc=5V
Hysterese Schmitt-triggers	6%	12%	Vcc	0.3 ... 0.6V bij Vcc=5V
Cext uitgang		0.4	V	I=2mA
Cext uitgang		2.0	V	I=5mA

## Voeding uit de USB bus

De USB bus kan voeding leveren voor de I/O Solution, en daarnaast voor eventuele extra hardware mits de totale stroombehoefte niet te groot is. De I/O Solution reserveert 100mA stroom uit de USB bus. Daarvan zijn 25mA voor de I/O Solution nodig, dus er blijven maximaal 75mA over voor additionele hardware.

## Suspend

Als een PC in stand-by staat, moeten alle apparaten die erop zijn aangesloten, ook in een energiezuinige toestand staan. Deze 'suspend' mode wordt door de I/O Solution herkend doordat de USB bus geen dataverkeer meer heeft. De I/O Solution gaat dan zelf in een zuinige toestand, maar de eventuele extra hardware die ook zijn voeding uit de USB bus betreft, moet officieel ook in een low-power toestand gaan. De CEXT pin van de I/O Solution kan worden gebruikt om externe hardware aan- en uit te schakelen. CEXT is een open-drain uitgang. Als deze uitgang 'laag' is, is de USB bus in vol bedrijf en mag de externe hardware de gereserveerde 75mA gebruiken (maximaal). Als de uitgang open is, moet de externe hardware uitgeschakeld worden en mag geen stroom meer gebruiken.

Pas op: Als de PC uit stand-by komt, kan de CEXT uitgang eerst een paar keer kortstondig laag worden, alvorens definitief laag te blijven.

## Communicatie met de PC

De I/O Solution communiceert met de PC via blokken van elk 8 bytes. De blokken hebben de volgende betekenis:

Byte	Vanuit PC	Betekenis
1	Schrijven	Stroominstelling P0.3 (4 meest significante bits) en P0.2 (4 minst significante bits)
2	Schrijven	Stroominstelling P0.1 (4 meest significante bits) en P0.0 (4 minst significante bits)
3	Schrijven	Stroominstelling P1.3 (4 meest significante bits) en P1.2 (4 minst significante bits)
4	Schrijven	Stroominstelling P1.1 (4 meest significante bits) en P1.0 (4 minst significante bits)
5	Schrijven	Instelling pull-up weerstanden poort 1 ( een '0' betekent de pull-up inschakelen) <sup>1</sup>
6	Schrijven	Instelling pull-up weerstanden poort 0 ( een '0' betekent de pull-up inschakelen)
7	Schrijven	Data voor poort 1 <sup>1</sup>
8	Schrijven	Data voor poort 0
1	Lezen	Data van poort 1
2	Lezen	Data van poort 0
3	Lezen	Data van poort 1
4	Lezen	Data van poort 0
5	Lezen	Data van poort 1
6	Lezen	Data van poort 0
7	Lezen	Data van poort 1
8	Lezen	Data van poort 0

1: De 4 meest significante bits van poort 1 zijn niet naar buiten uitgevoerd. Ze zijn intern echter wel aanwezig. Het is het beste om voor deze pinnen de pull-up te activeren, en dus de 4 meest significantte bits in byte4 een '0' te maken.

Als er data binnenkomt verwerkt de I/O solution deze in de volgorde waarin de bytes staan in de bovenstaande tabel. De volgorde van verwerken is dus:

1. De stroom voor P0.3 wordt ingesteld
2. De stroom voor P0.2 wordt ingesteld
3. De stroom voor P0.1 wordt ingesteld
4. De stroom voor P0.0 wordt ingesteld
5. De stroom voor P1.3 wordt ingesteld

6. De stroom voor P1.2 wordt ingesteld
7. De stroom voor P1.1 wordt ingesteld
8. De stroom voor P1.0 wordt ingesteld
9. De pull-ups voor poort 1 worden ingesteld
10. De pull-ups voor poort 0 worden ingesteld
11. De data voor poort 1 wordt geschreven
12. De data voor poort 0 wordt geschreven

Tussen elke stap zit een aantal micro-seconden. Voor de meeste toepassingen zal het niet nodig zijn daar rekening mee te houden.

Als er data opgevraagd wordt leest de I/O solution de niveaus op de I/O pinnen vier maal snel achter elkaar:

1. Poort 1 wordt gelezen
2. Poort 0 wordt gelezen
3. Poort 1 wordt gelezen
4. Poort 0 wordt gelezen
5. Poort 1 wordt gelezen
6. Poort 0 wordt gelezen
7. Poort 1 wordt gelezen
8. Poort 0 wordt gelezen

Tussen de lees acties zit steeds een paar micro-seconden. Als het ingangs signaal binnen die tijd niet stabiel is, bijvoorbeeld omdat een pin juist bezig is van toestand te veranderen, is dit dus terug te zien in de data die gelezen wordt. Voor de meeste toepassingen zal het niet nodig zijn daar rekening mee te houden, en zijn alleen de eerste twee bytes interessant.

## Software

Vanuit een eigen programma kan de I/O Solution worden bereikt door middel van de standaard CreateFile, ReadFile en WriteFile routines. In Delphi:

```
var
  HidDevice: THandle;
  BytesWritten: DWORD;
  Success: Boolean;
  Buffer: Array[0..8] of Byte;
  BufSize: Integer;

// handle naar het device openen
HidDevice := CreateFile( DeviceName,
                       GENERIC_READ or GENERIC_WRITE,
                       FILE_SHARE_READ or FILE_SHARE_WRITE,
                       Nil, OPEN_EXISTING, 0, 0);

// handle gebruiken voor schrijf actie
Buffer[0] := 0; // Dit altijd op 0 zetten
Buffer[1] := $FF; // Instellen stroom P03 en P02
Buffer[2] := $FF; // Instellen stroom P01 en P00
Buffer[3] := $FF; // Instellen stroom P13 en P12
Buffer[4] := $FF; // Instellen stroom P11 en P10
Buffer[5] := $00; // Instellen pull-ups poort 1
Buffer[6] := $00; // Instellen pull-ups poort 0
Buffer[7] := $00; // Data voor poort 1
Buffer[8] := $00; // Data voor poort 0

Success := WriteFile( FHandle, Buffer, SizeOf( Buffer), BytesWritten, Nil);

// handle gebruiken voor lees actie
BufSize := SizeOf( Buffer);
ReadFile( HidDevice, Buffer, BufSize, BytesRead, Nil);

// Buffer[1] bevat nu de data die werd gelezen van poort 1
// Buffer[2] bevat nu de data die werd gelezen van poort 0

// handle weer sluiten
CloseHandle( HidDevice);
```

In Visual Basic of C++ zijn soortgelijke routines te gebruiken. Uiteraard is dit maar summere code, waar nog de nodige fout afhandeling ed. aan dient te worden toegevoegd. Ook is de naam van het device bekend verondersteld. Die naam wordt echter door Windows aan het device gegeven op het moment dat dit wordt aangesloten. Dat is dus een steeds veranderend gegeven, en de naam die gebruikt moet worden moet dan ook eerst worden opgezocht in de lijst met aangesloten devices, die door Windows wordt bijgehouden. Een voert te ver om hier te behandelen, maar voldoende informatie hierover kunt u vinden op de website: [www.elomax.nl](http://www.elomax.nl) Daar zijn ook kant- en klare componenten te vinden voor Delphi, die het programmeren van de I/O Solution vanuit Delphi of C++ Builder sterk vereenvoudigen.

