Physical Computing

2007-2008

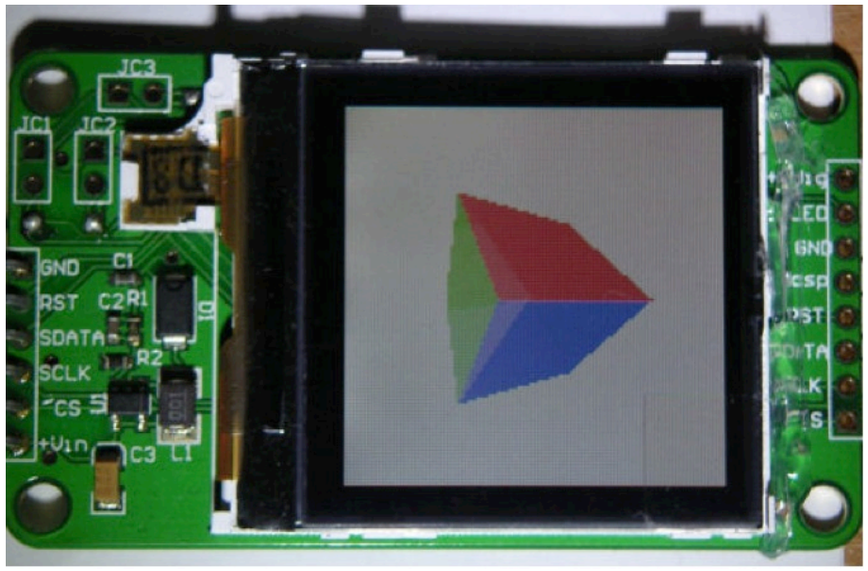
Interpretatie I

Programmeerproject I

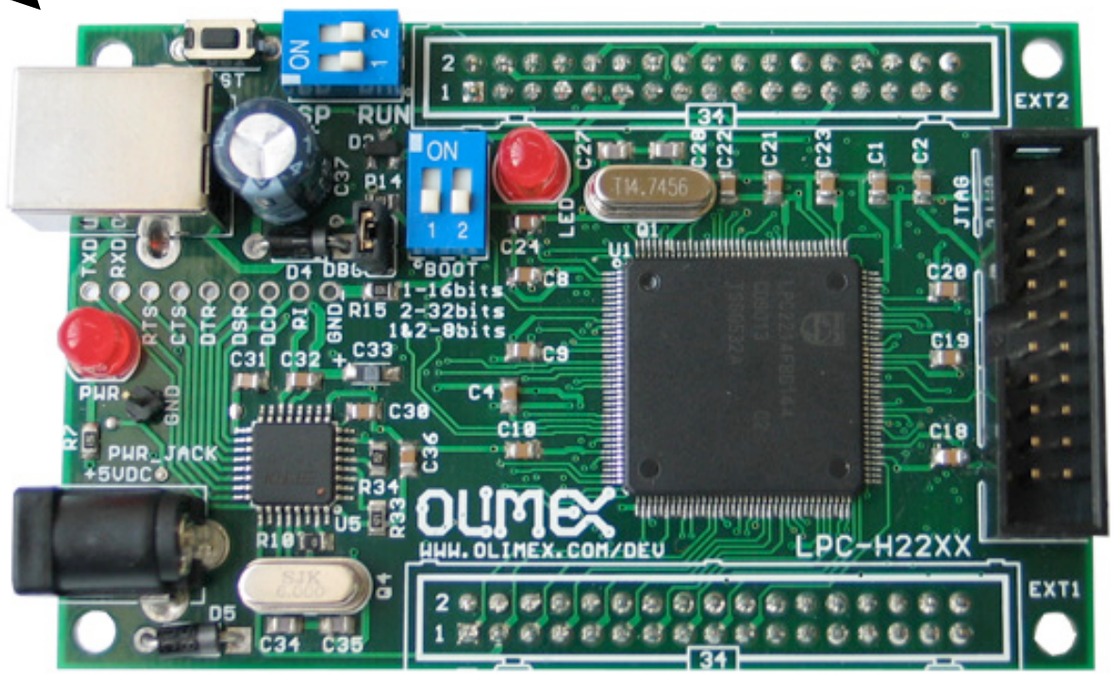
Coen De Roover - Christophe Scholliers - Yves Vandriessche

Programming Technology Lab
Vrije Universiteit Brussel
Belgium

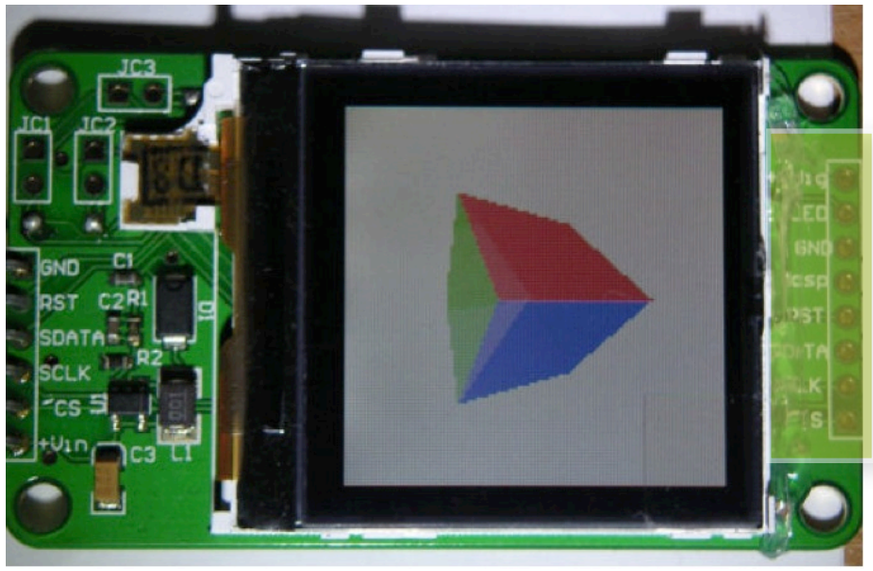
LCD Scherm


IC3
IC1
IC2
GND
C1
RST
C2
R1
SDATA
R2
SCLK
CS
+Vin
C3
L1

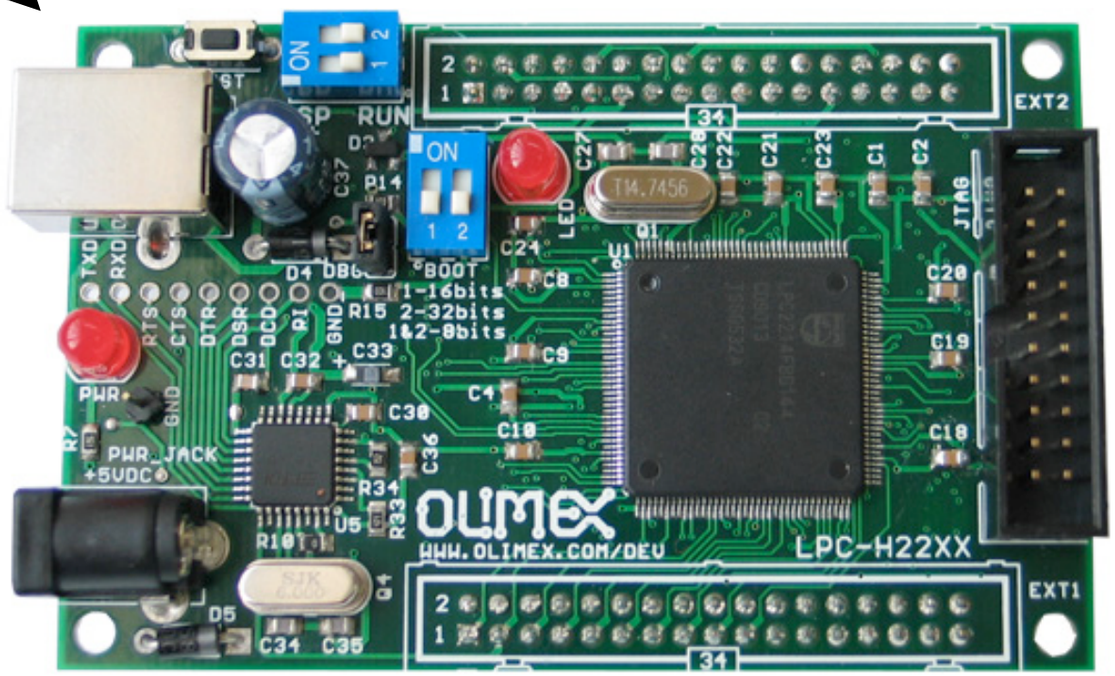
? ? ?


ON
1 2
ST
SP
RUN
ON
1 2
BOOT
1-16bits
R15 2-32bits
1&2-8bits
LED
PWR
PWR_JACK
+5VDC
OLIMEX
WWW.OLIMEX.COM/DEV
LPC-H22XX
EXT2
EXT1
JTAG

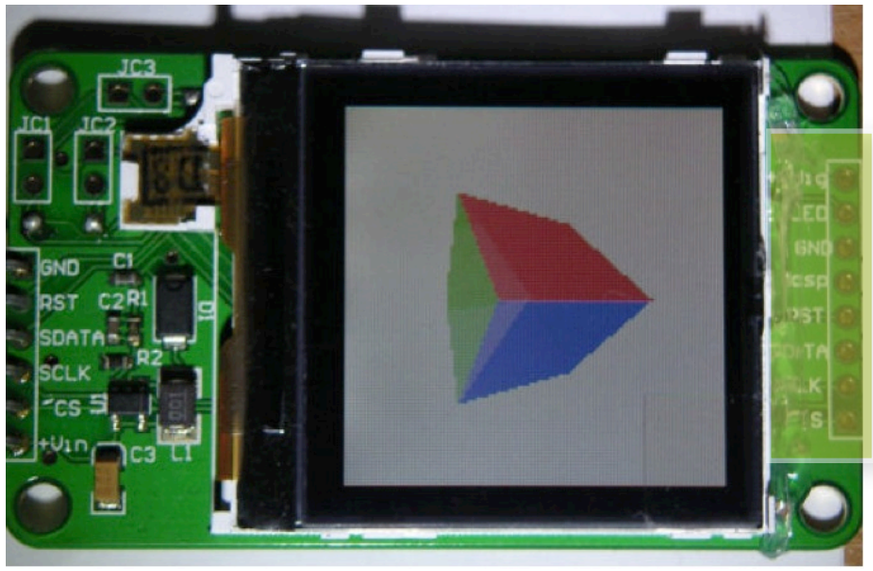
LCD Scherm


IC3
IC1
IC2
GND
C1
RST
C2
R1
SDATA
R2
SCLK
CS
+Vin
C3
L1

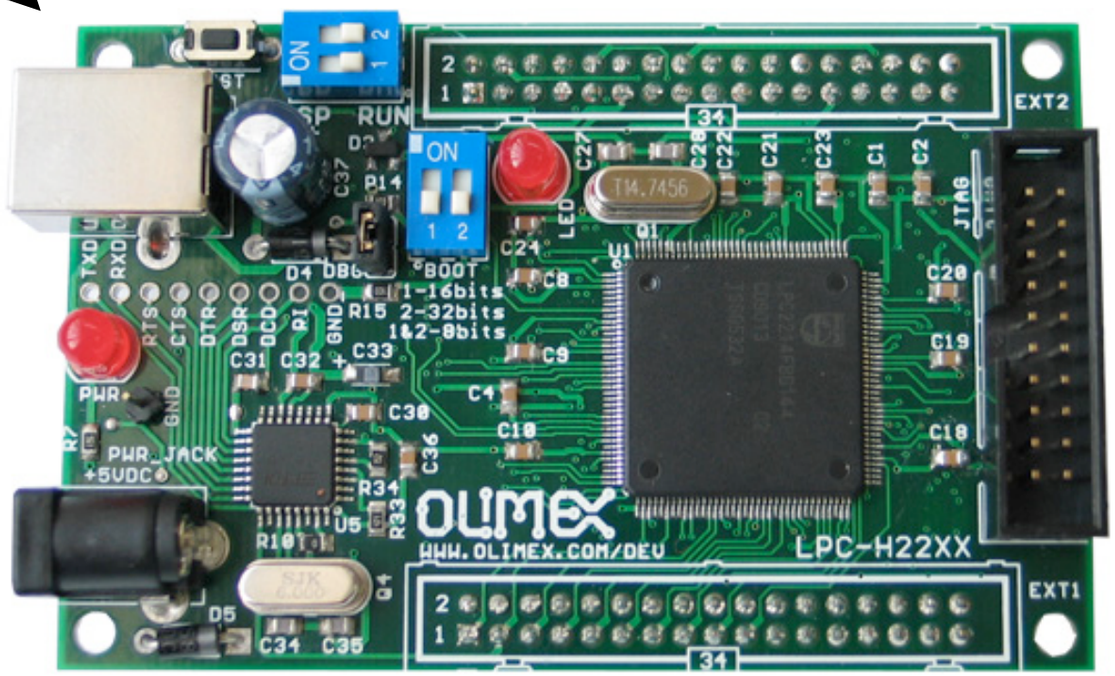
? ? ?


ON
1 2
ST
SP
RUN
D3
P14
C37
ON
1 2
BOOT
LED
C27
C26
C22
C21
C23
C1
C2
JTAG
EXT2
34
T14.7456
Q1
U1
C24
C8
C20
C19
C18
C18
TXD
RXD
RTS
CTS
DTR
DSR
DCD
RI
GND
D4 DB6
1-16bits
R15 2-32bits
1&2-8bits
C31
C32
C33
C9
C4
C30
C36
PWR
GND
R7
PWR_JACK
+5VDC
R34
R33
U5
R10
OLIMEX
WWW.OLIMEX.COM/DEV
LPC-H22XX
Q4
D5
C34
C35
EXT1
34

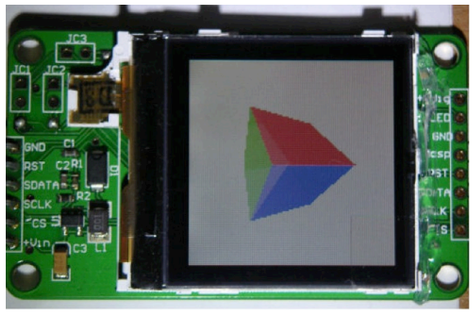
LCD Scherm



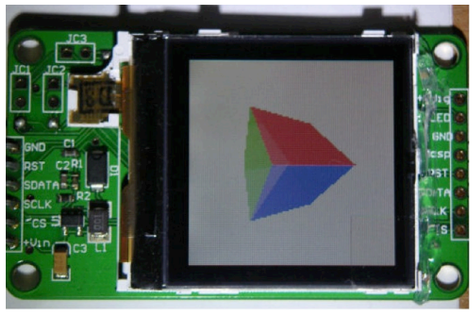
Serial Protocol Interface (SPI)



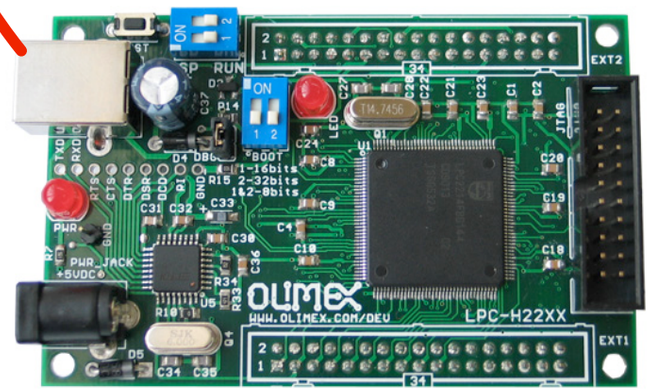
Serial Protocol Interface



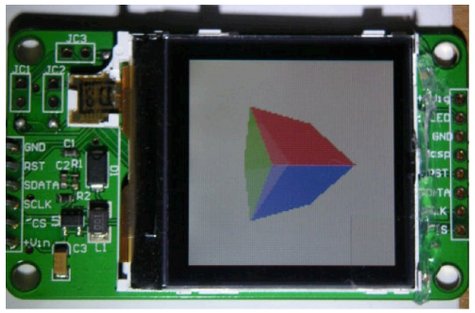
Serial Protocol Interface



sdata

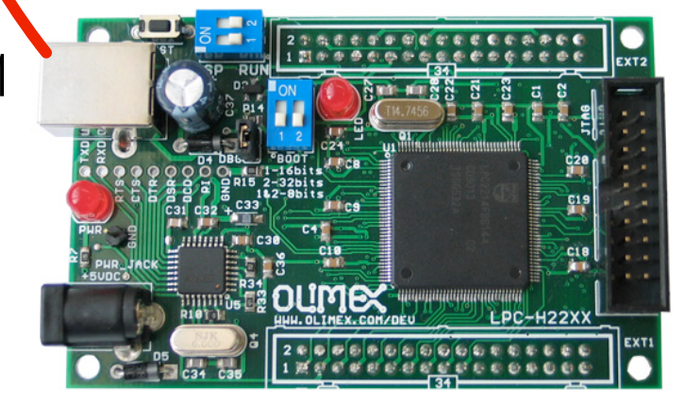


Serial Protocol Interface

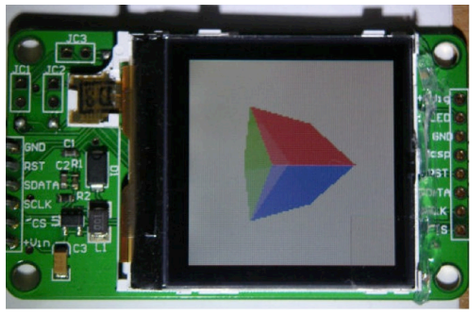


sdata

1 1 0 1



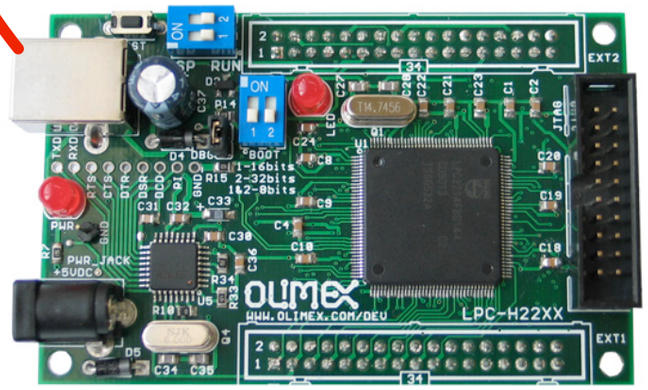
Serial Protocol Interface



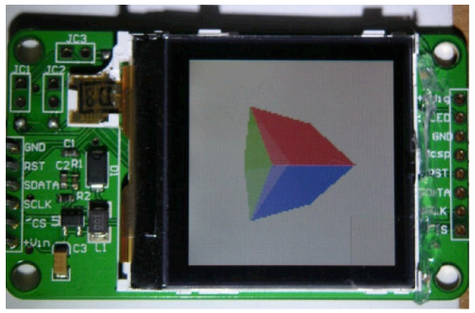
??

sdata

1 1 0 1



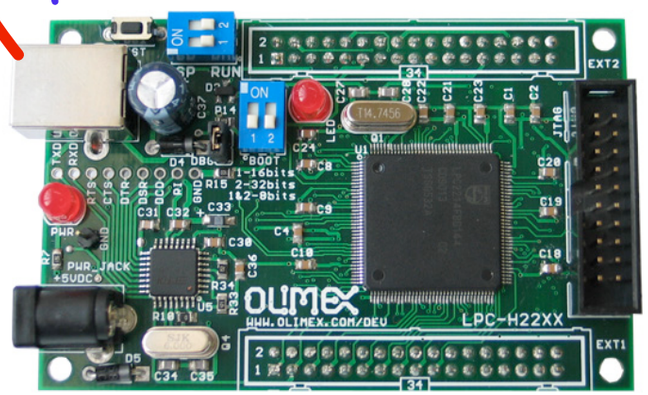
Serial Protocol Interface



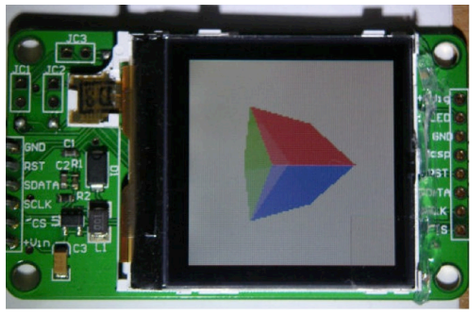
??

sdata

1 1 0 1



Serial Protocol Interface

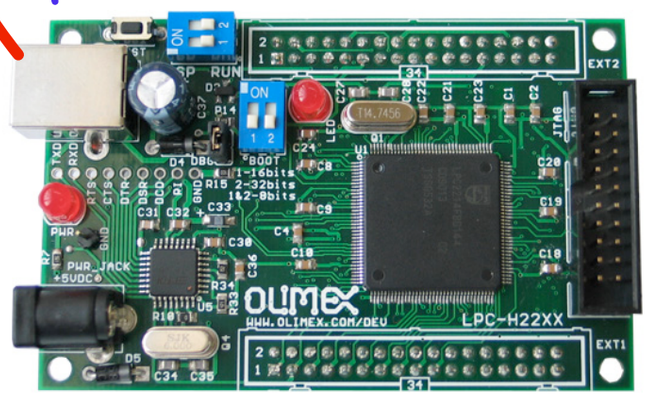


??

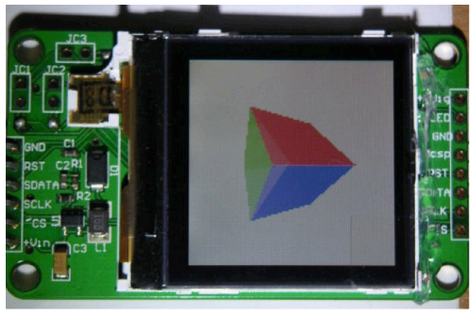
sclk

sdata

1 1 0 1



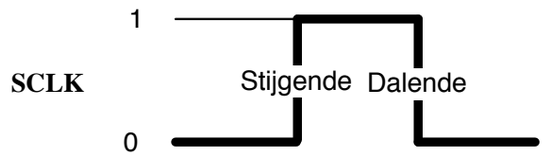
Serial Protocol Interface



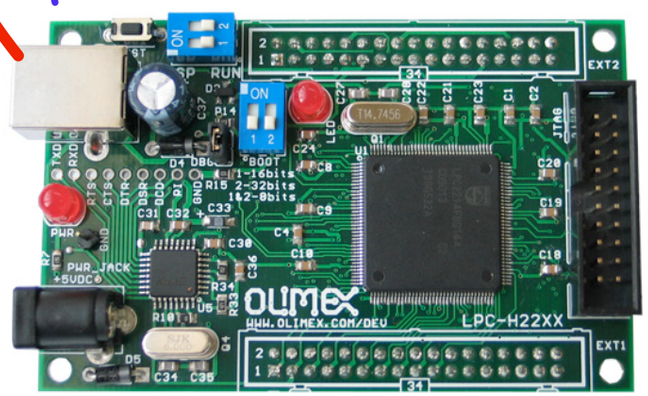
??

sclk

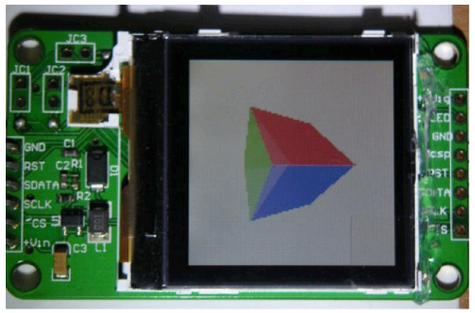
sdata



1 1 0 1



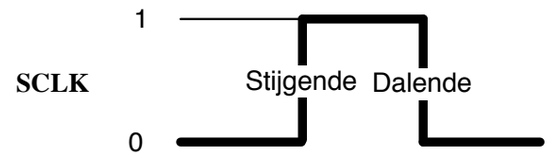
Serial Protocol Interface



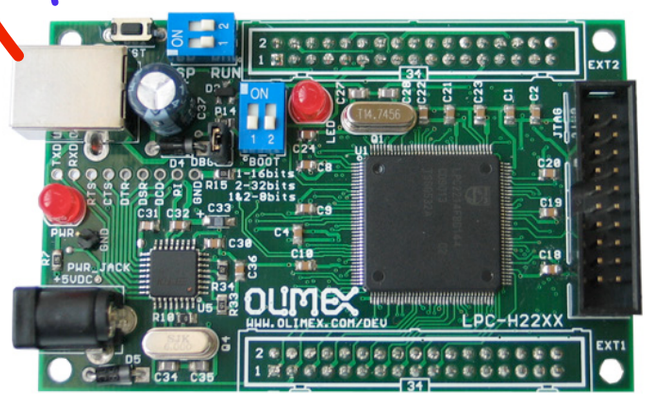
??

sclk

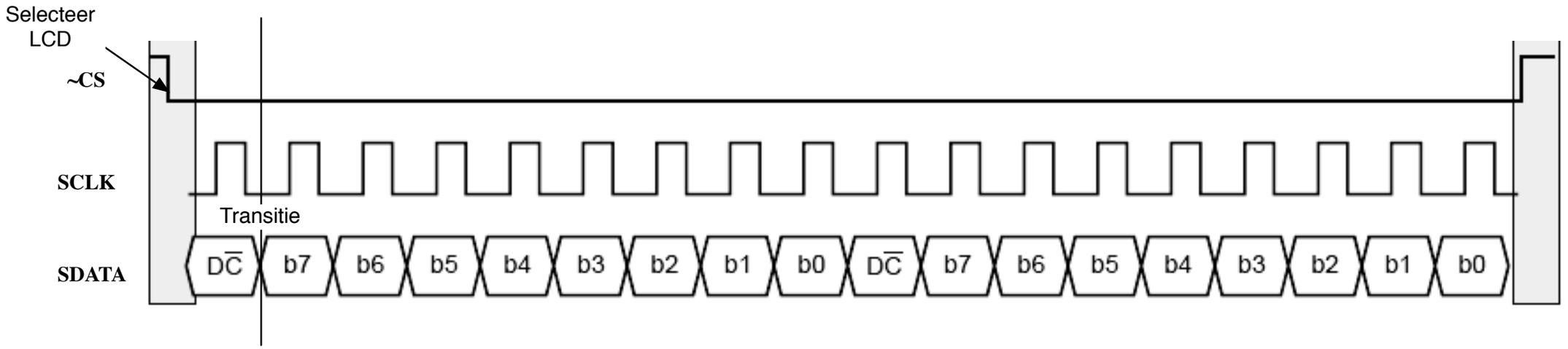
sdata



0
1
0
1
0
1
1
0
1

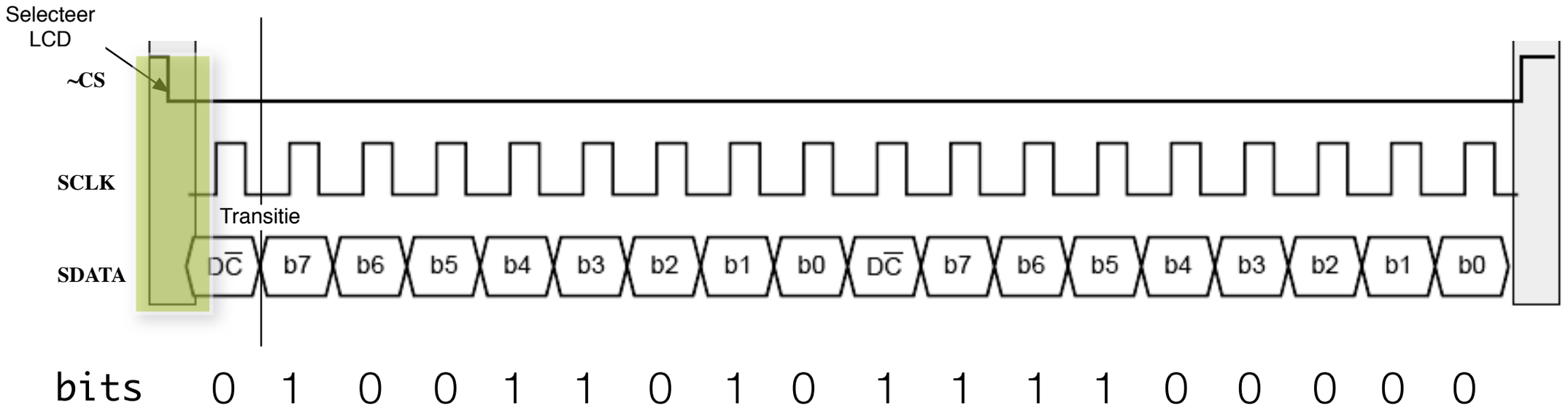


Serial Protocol Interface



bits 0 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0

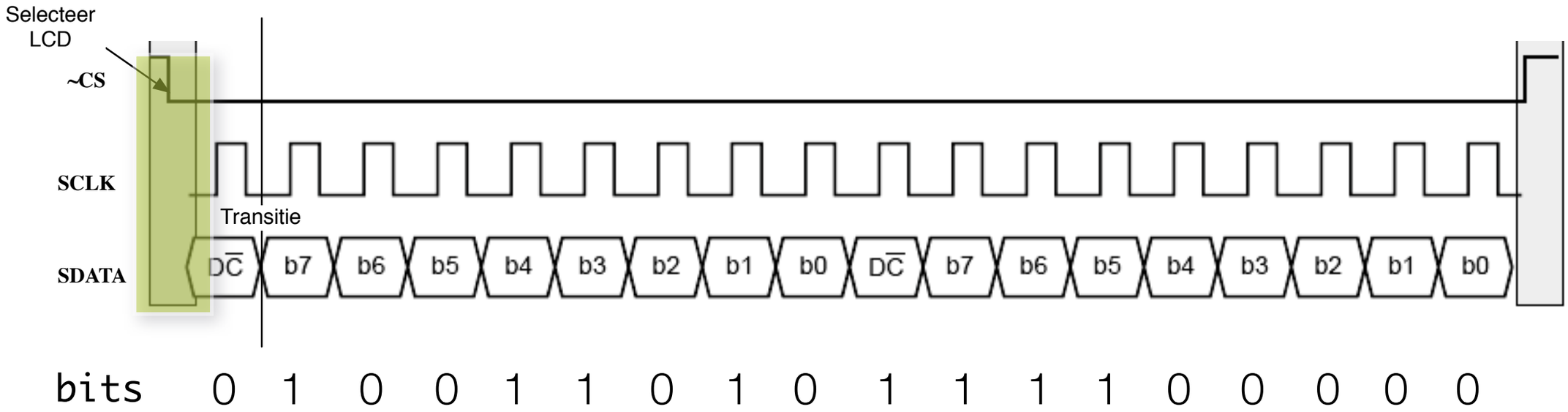
Serial Protocol Interface



0

1. Cable Select low
2. Send Instructions
3. Cable Select high

Serial Protocol Interface

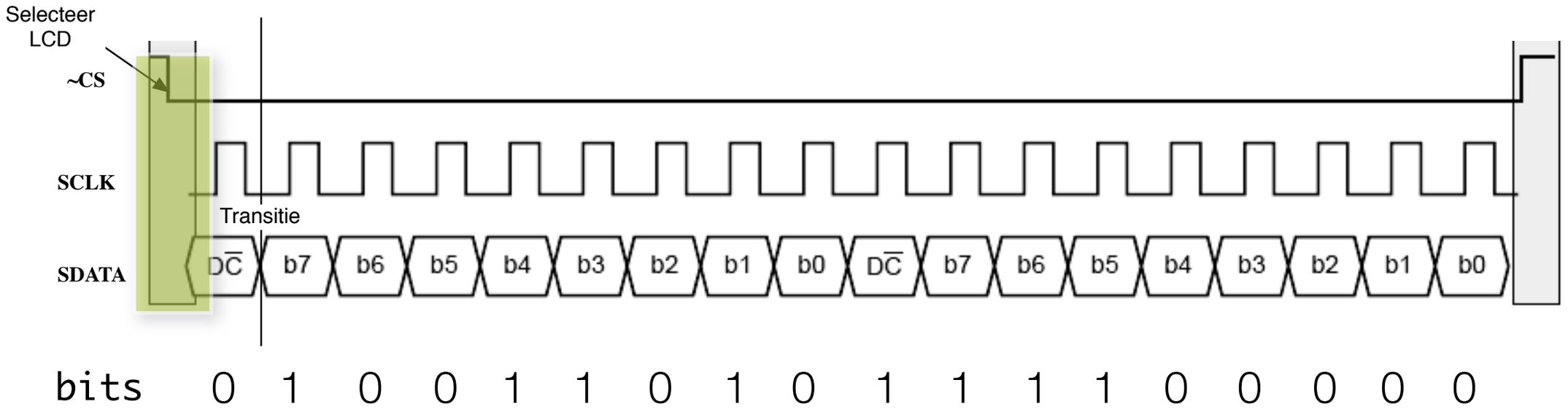


0

1. Cable Select low
2. Send Instructions
   a. Send Command bit
   b. Send Data bits
3. Cable Select high

```
(define (send-byte cmd data)
        (clear-pin cs)
        ;; send the command bit
        (send-bit cmd)
        ;; send the byte
        (let loop ((bitnr 7))
          (if (>= bitnr 0)
           (begin
             (send-bit (get-bit data bitnr))
             (loop (- bitnr 1)))))
        (set-pin cs))
```

Serial Protocol Interface

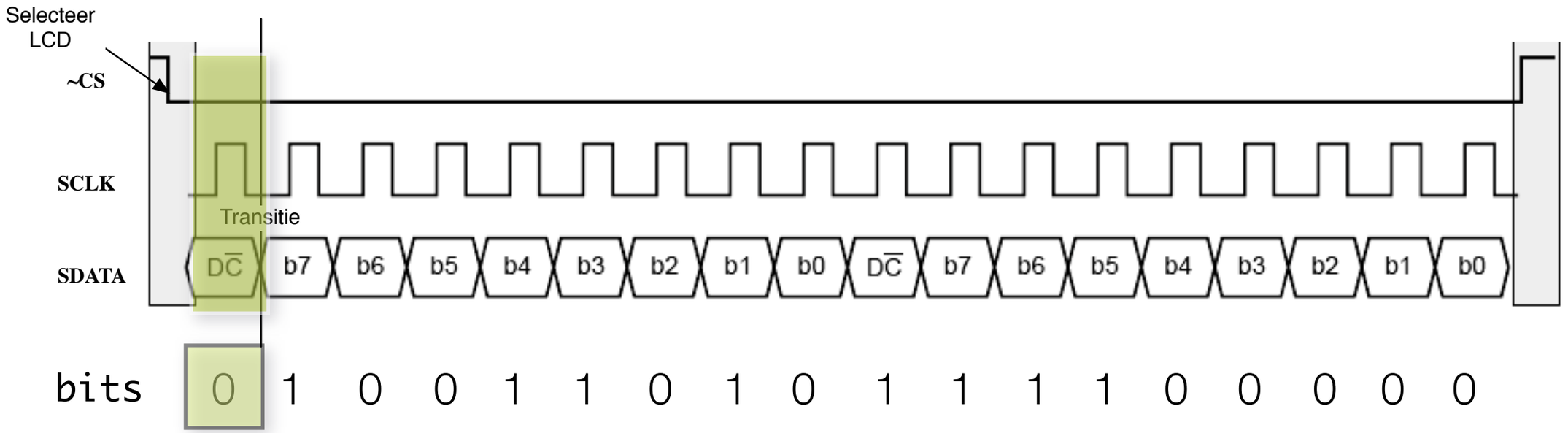


0

1. Cable Select low
2. Send Instructions
   a. Send Command bit
   b. Send Data bits
3. Cable Select high

```
(define (send-byte cmd data)
        (clear-pin cs)
        ;; send the command bit
        (send-bit cmd)
        ;; send the byte
        (let loop ((bitnr 7))
          (if (>= bitnr 0)
           (begin
             (send-bit (get-bit data bitnr))
             (loop (- bitnr 1)))))
        (set-pin cs))
```

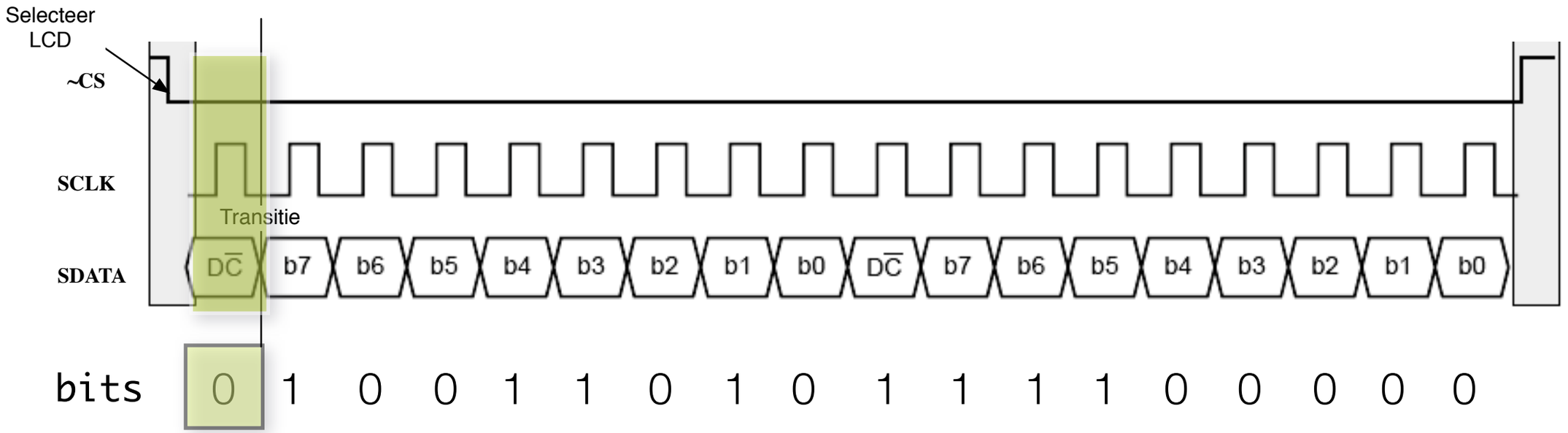
Serial Protocol Interface



1. Cable Select low
2. Send Instructions
   a. Send Command bit
   b. Send Data bits
3. Cable Select high

```
(define (send-byte cmd data)
        (clear-pin cs)
        ;; send the command bit
        (send-bit cmd)
        ;; send the byte
        (let loop ((bitnr 7))
          (if (>= bitnr 0)
           (begin
             (send-bit (get-bit data bitnr))
             (loop (- bitnr 1)))))
        (set-pin cs))
```

Serial Protocol Interface

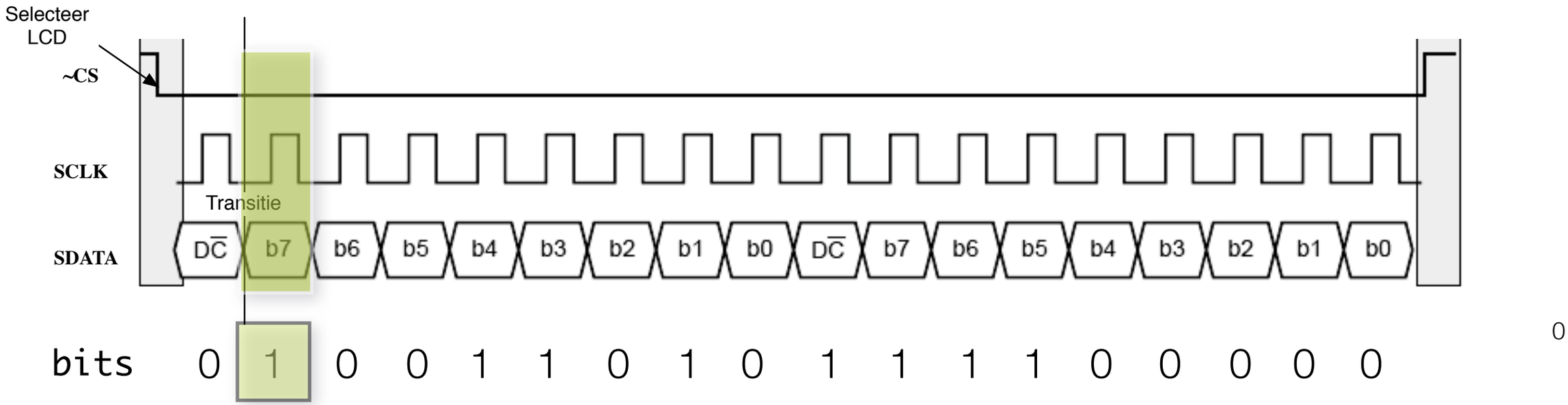


1. Cable Select low
2. Send Instructions
   a. Send Command bit
   b. Send Data bits
3. Cable Select high

```
(define (send-byte cmd data)
        (clear-pin cs)
        ;; send the command bit
        (send-bit cmd)
        ;; send the byte
        (let loop ((bitnr 7))
          (if (>= bitnr 0)
           (begin
             (send-bit (get-bit data bitnr))
             (loop (- bitnr 1)))))
        (set-pin cs))
```

```
(define (send-bit bit)
  (clear-pin sclk)
  (if (eq? bit 0)
    (clear-pin sdata)
    (set-pin sdata))
  (set-pin sclk))
```

Serial Protocol Interface

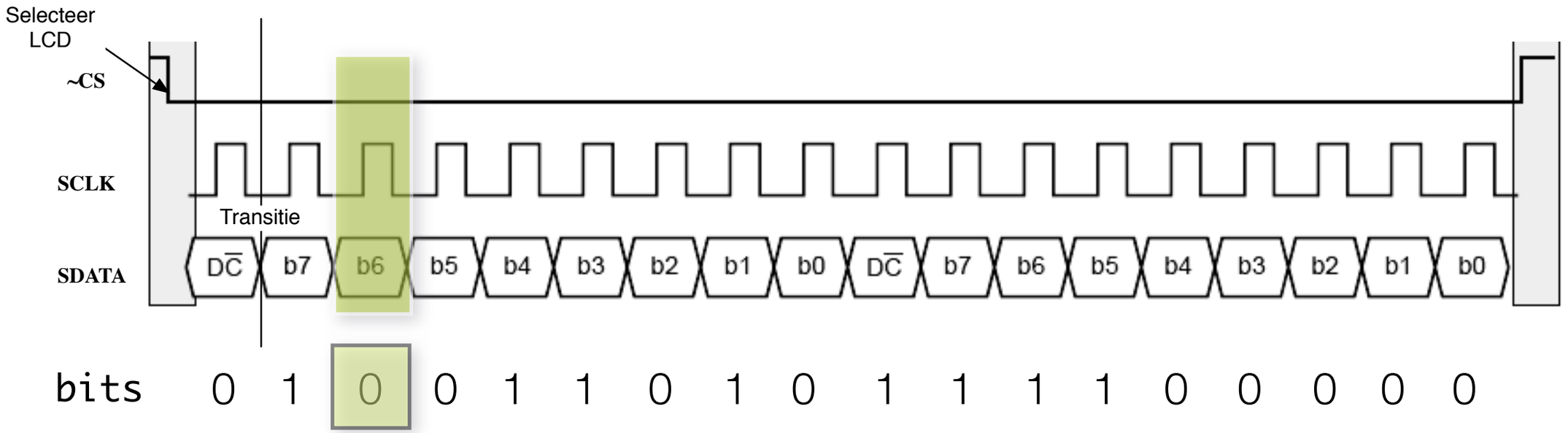


1. Cable Select low
2. Send Instructions
   a. Send Command bit
   b. Send Data bits
3. Cable Select high

```
(define (send-byte cmd data)
        (clear-pin cs)
        ;; send the command bit
        (send-bit cmd)
        ;; send the byte
        (let loop ((bitnr 7))
          (if (>= bitnr 0)
           (begin
             (send-bit (get-bit data bitnr))
             (loop (- bitnr 1)))))
        (set-pin cs))
```

```
(define (send-bit bit)
  (clear-pin sclk)
  (if (eq? bit 0)
    (clear-pin sdata)
    (set-pin sdata))
  (set-pin sclk))
```

Serial Protocol Interface



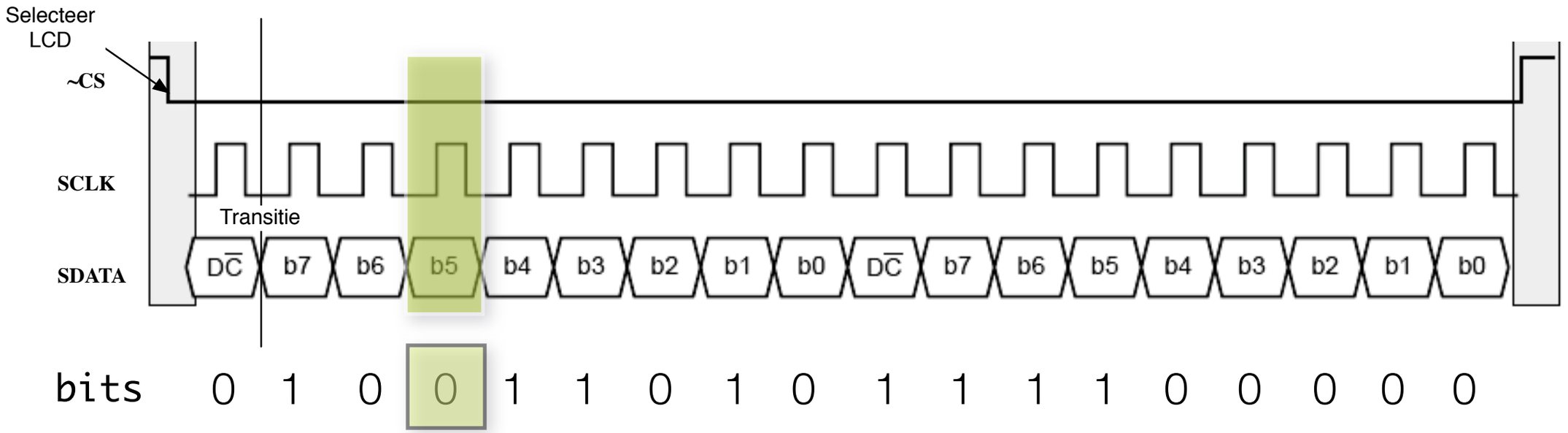
0

1. Cable Select low
2. Send Instructions
   a. Send Command bit
   b. Send Data bits
3. Cable Select high

```
(define (send-byte cmd data)
        (clear-pin cs)
        ;; send the command bit
        (send-bit cmd)
        ;; send the byte
        (let loop ((bitnr 7))
          (if (>= bitnr 0)
          (begin
            (send-bit (get-bit data bitnr))
            (loop (- bitnr 1)))))
        (set-pin cs))
```

```
(define (send-bit bit)
  (clear-pin sclk)
  (if (eq? bit 0)
    (clear-pin sdata)
    (set-pin sdata))
  (set-pin sclk))
```

Serial Protocol Interface

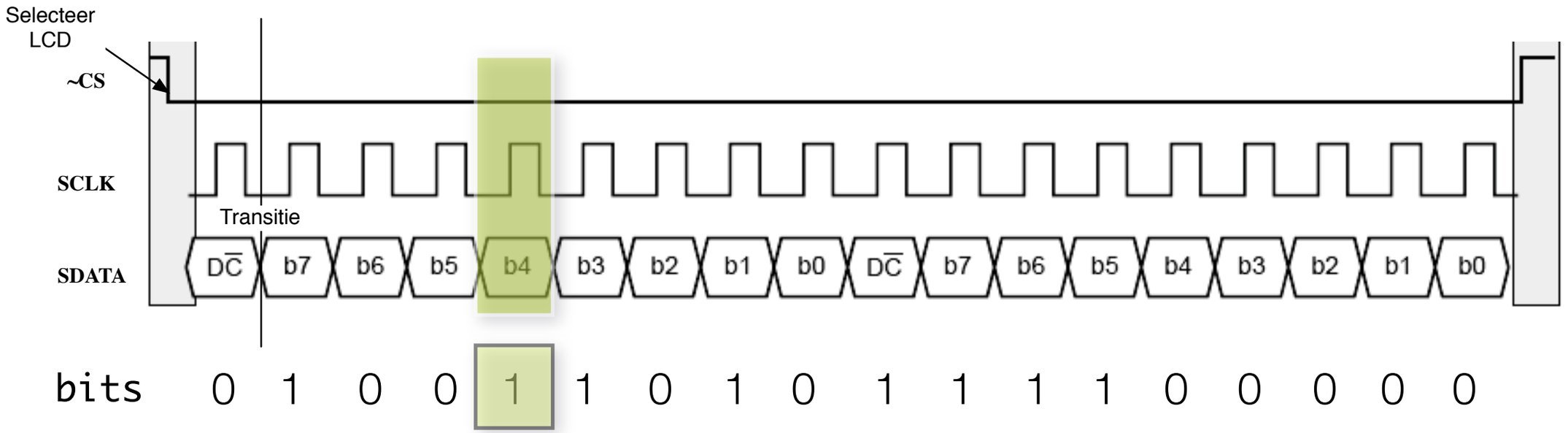


1. Cable Select low
2. Send Instructions
   a. Send Command bit
   b. Send Data bits
3. Cable Select high

```
(define (send-byte cmd data)
        (clear-pin cs)
        ;; send the command bit
        (send-bit cmd)
        ;; send the byte
        (let loop ((bitnr 7))
          (if (>= bitnr 0)
          (begin
            (send-bit (get-bit data bitnr))
            (loop (- bitnr 1)))))
        (set-pin cs))
```

```
(define (send-bit bit)
  (clear-pin sclk)
  (if (eq? bit 0)
    (clear-pin sdata)
    (set-pin sdata))
  (set-pin sclk))
```

Serial Protocol Interface

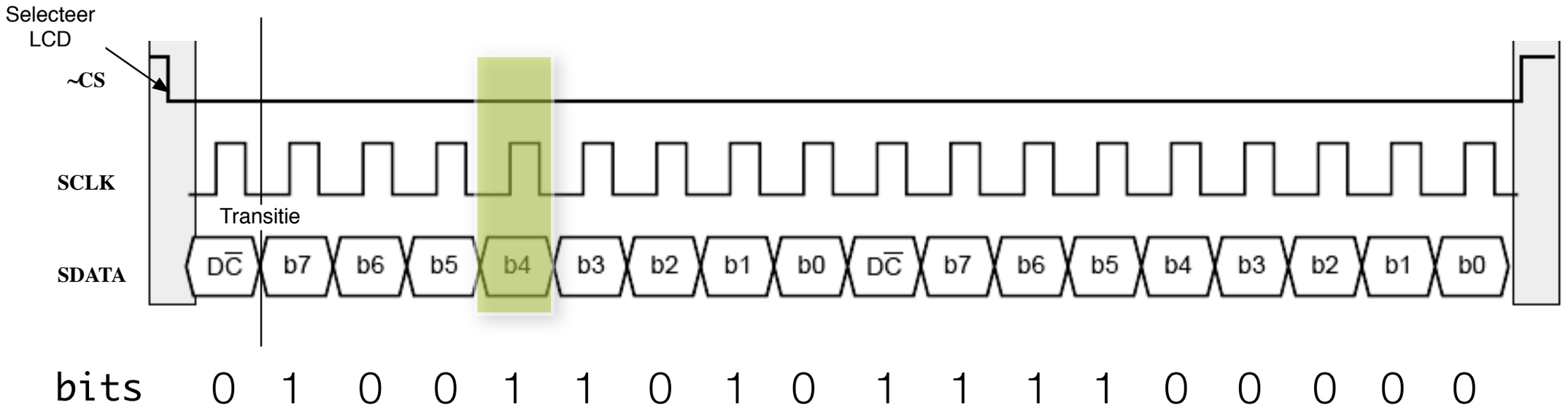


1. Cable Select low
2. Send Instructions
   a. Send Command bit
   b. Send Data bits
3. Cable Select high

```
(define (send-byte cmd data)
        (clear-pin cs)
        ;; send the command bit
        (send-bit cmd)
        ;; send the byte
        (let loop ((bitnr 7))
          (if (>= bitnr 0)
          (begin
            (send-bit (get-bit data bitnr))
            (loop (- bitnr 1)))))
        (set-pin cs))
```

```
(define (send-bit bit)
  (clear-pin sclk)
  (if (eq? bit 0)
    (clear-pin sdata)
    (set-pin sdata))
  (set-pin sclk))
```

Serial Protocol Interface



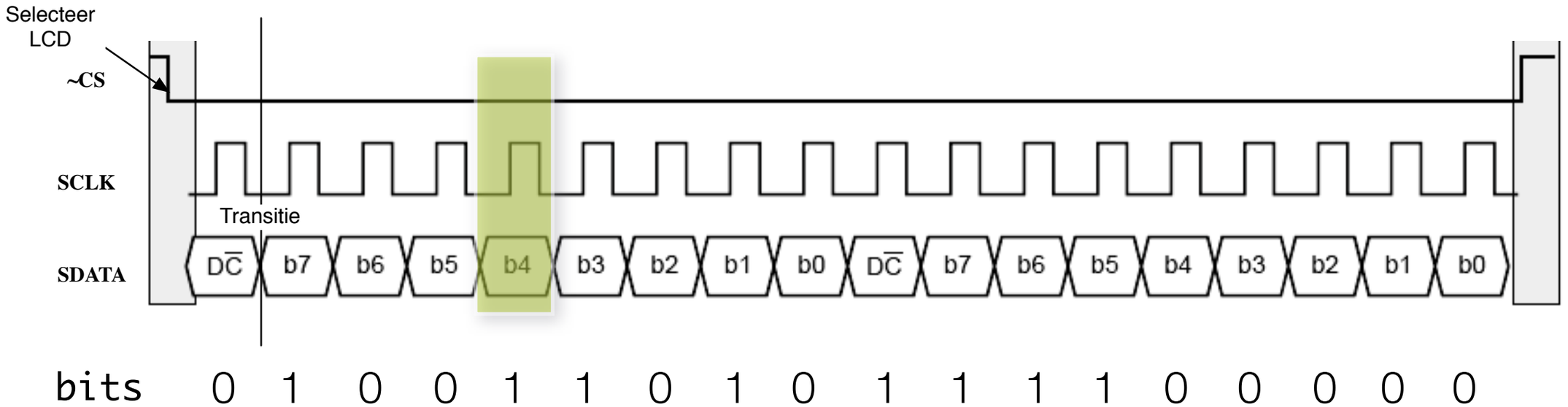
0

1. Cable Select low
2. Send Instructions
   a. Send Command bit
   b. Send Data bits
3. Cable Select high

```
(define (send-byte cmd data)
        (clear-pin cs)
        ;; send the command bit
        (send-bit cmd)
        ;; send the byte
        (let loop ((bitnr 7))
          (if (>= bitnr 0)
           (begin
             (send-bit (get-bit data bitnr))
             (loop (- bitnr 1)))))
        (set-pin cs))
```

```
(define (send-bit bit)
  (clear-pin sclk)
  (if (eq? bit 0)
    (clear-pin sdata)
    (set-pin sdata))
  (set-pin sclk))
```

Serial Protocol Interface

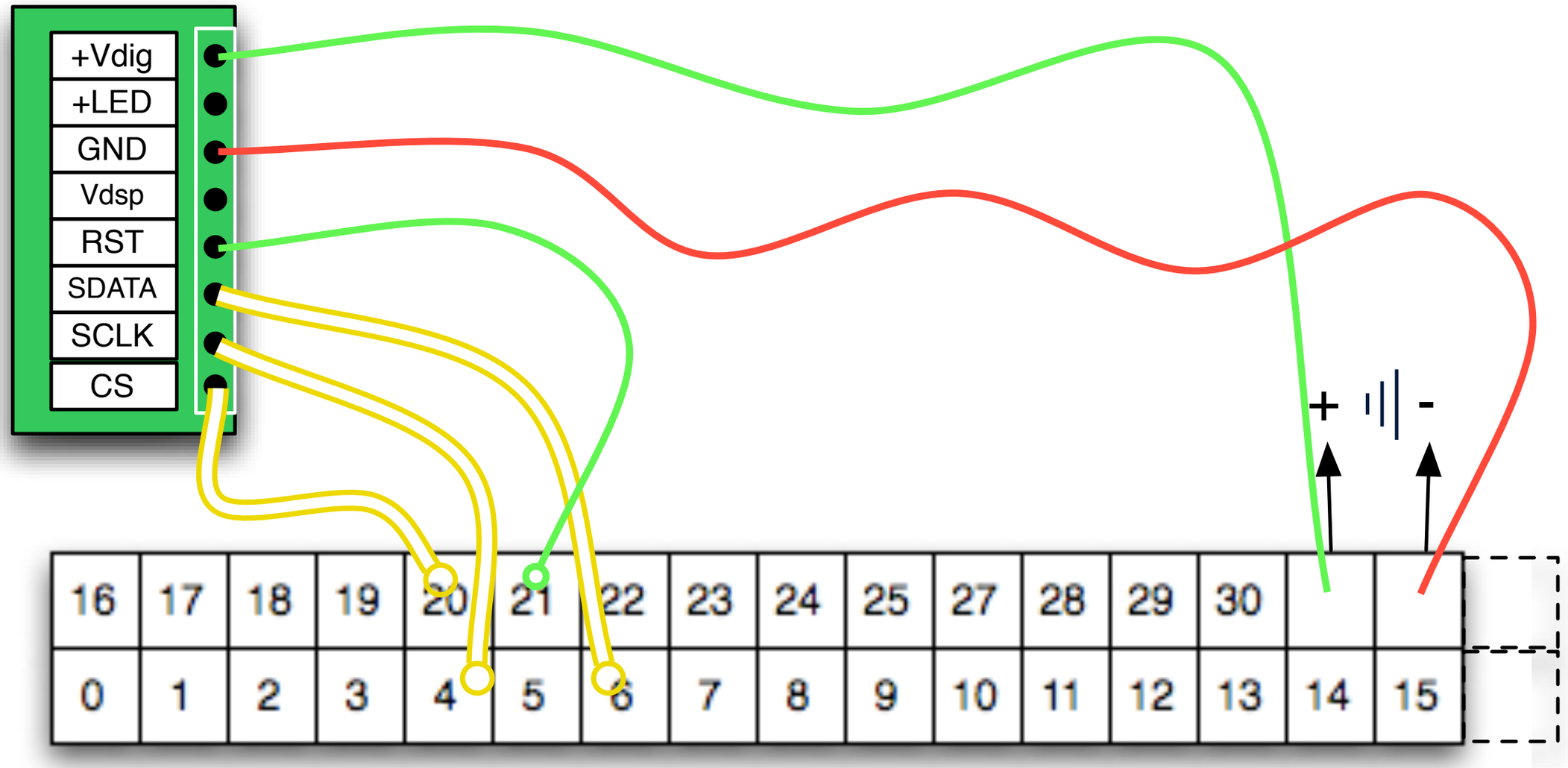


1. Cable Select low
2. Send Instructions
   a. Send Command bit
   b. Send Data bits
3. Cable Select high

```
(define (send-byte cmd data)
        (clear-pin cs)
        ;; send the command bit
        (send-bit cmd)
        ;; send the byte
        (let loop ((bitnr 7))
          (if (>= bitnr 0)
           (begin
             (send-bit (get-bit data bitnr))
             (loop (- bitnr 1)))))
        (set-pin cs))
```

```
(define (send-bit bit)
  (clear-pin sclk)
  (if (eq? bit 0)
    (clear-pin sdata)
    (set-pin sdata))
  (set-pin sclk))
```

Pin Scheme


+Vdig
+LED
GND
Vdsp
RST
SDATA
SCLK
CS
+
-
16 17 18 19 20 21 22 23 24 25 27 28 29 30
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15