TPS application sheet

people counting with

TPS 210

english

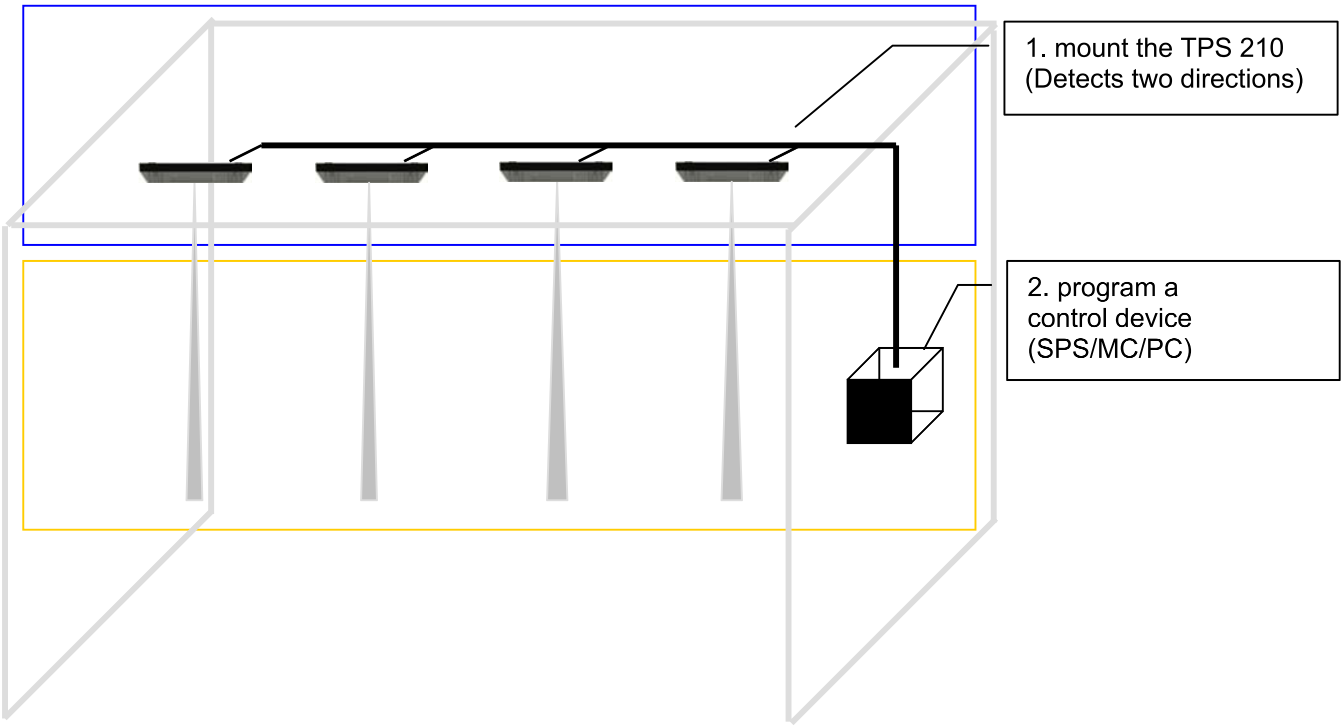
Example with 3 TPS 210

This application sheet can be used as basics for TPS210 people counting. The goal is to give our past knowledge about people counting with the TPS210.

Standard setup

A standard setup can be split in two sections:

1. mount the devices as described in the manual
2. program a control device



1. mount the TPS 210 (Detects two directions)

2. program a control device (SPS/MC/PC)

Filter

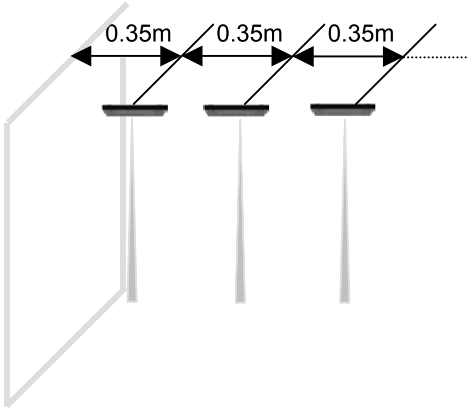
The TPS 210 device uses a filter to discard swinging arms or objects that disturb the people counting. The effect of this filter is, that objects with a length of 0.30m passing the light beams faster than 5m/s could not be recognized.

Mount the TPS 210

For proper applications the number of sensors is very important.

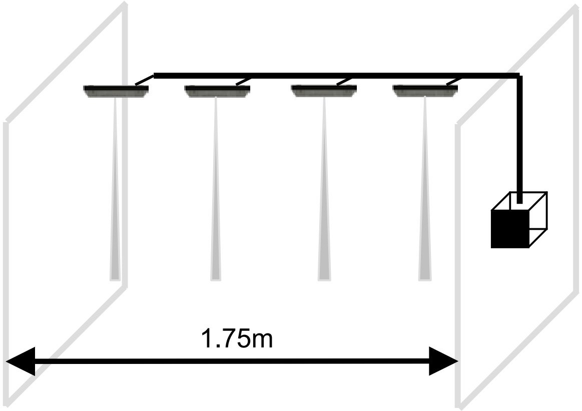
As told in the Operation Manual of the TPS 210, there is a minimal gap of 0.20m between tow TPS devices. With the 0.15m wide TPS 210 housing there is a light beam every 0.35m. For people counting these gap is enough to detect a passing person at least with one sensor. Increasing this gap can effect incorrect counting. Unfortunately at present there is no statistic evaluations of this effect. First field-tests are running and will be evaluated as soon as possible.

To detect a person at least with one sensor the following examples show the number of sensors needed in different environments.



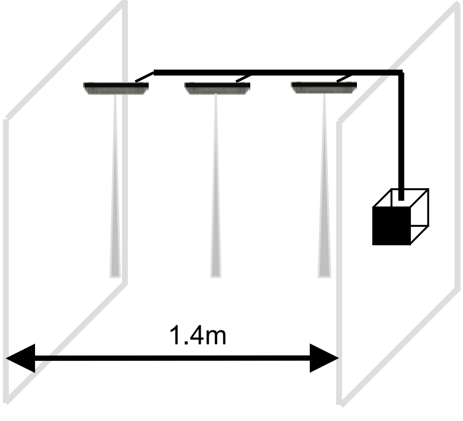
Nummber of TPS = (Gate width / Gap between ligth beams) - 1

Examples:



Gate of 1.75m :

nr. TPS = **(1.75m / 0.35m) – 1 = 4** x TPS 210



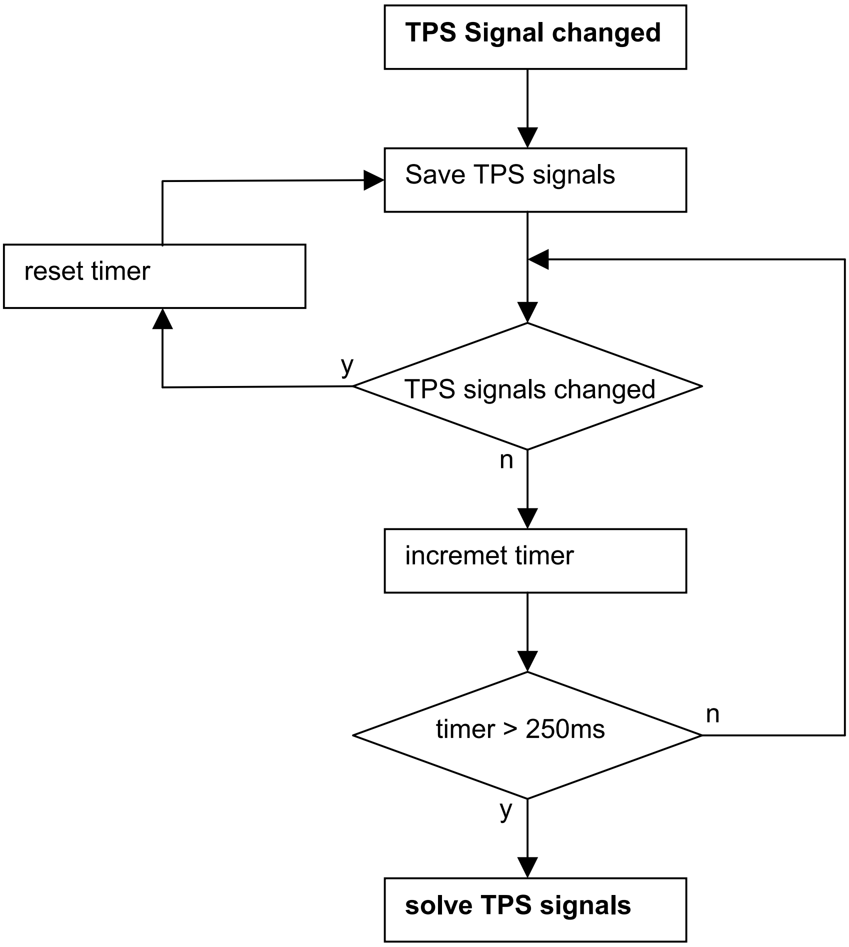
Gate of 1.4m :

nr. TPS = **(1.4m / 0.35m) – 1 = 3** x TPS 210

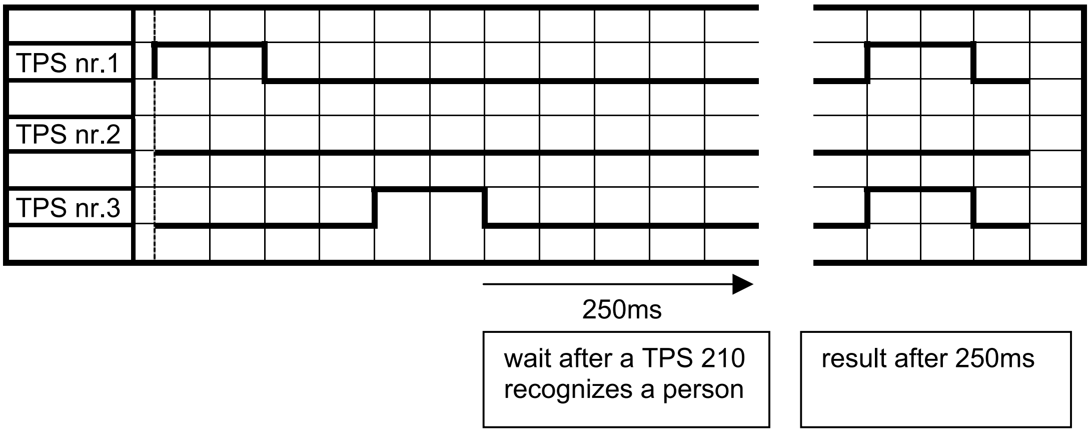
Control Device

If TPS device recognizes a passing person the control device has to wait 250ms to observe the neighbouring devices. (A passing person will often be recognized by two TPS, but not at the same time!)

Example for analyzing TPS 210 signals for people counting:



Example for analyzing TPS 210 signals.



Solve TPS signals with a table

Following table can solve the result for one direction (with three TPS210):

var	TPS nr.3	TPS nr.2	TPS nr.1	People count in one direction
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	1
6	1	0	1	2
7	1	1	0	1
8	1	1	1	2

Example for PIC controller in C:

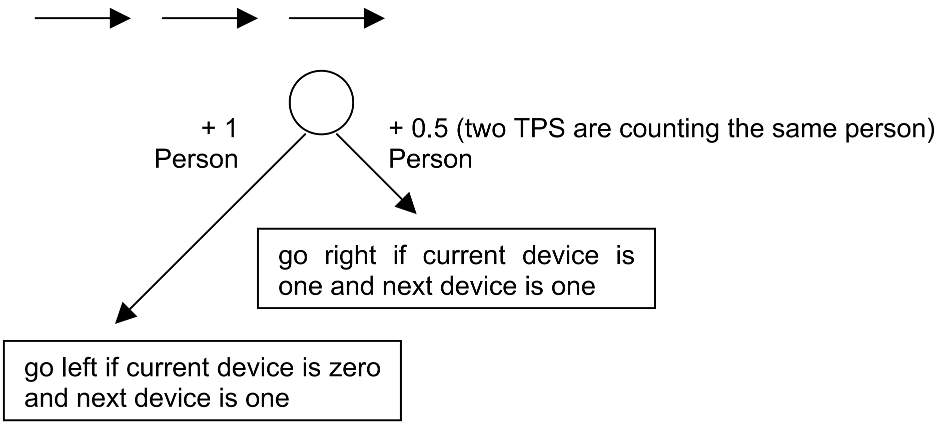
```
unsigned char In = 0b000;                      // variable for reading tps signals
unsigned char In_work =0b0000;                 // variable for storing tps signals
unsigned char TimerIn = 0b0000;                // variable for timer

while(1)
{
  ...
  In = TPSSignalIn();                          // read tps in signals

  if ( In > 0 )                                // save tps signals
  {
    In_work = In_work | In;                    // save changes to old tps signals
    TimerIn = 0;                               // reset timer
  }

  if (In_work > 0 )                            // increment timer if theres some information from the tps
  {
    TimerIn++;
  }

  if (TimerIn > 250MS)                         // solve the tps signals after 250ms silence on inputs
  {
      counterIn ++;                            // increment counterIn

      switch (In_work)                         // check the two cases (0b101/0b0111) for two persons counting
      {
        case 0b0100         : counterIn ++;    // 0b101 -> two persons counting
        break;
        case 0b0111         : counterIn ++;    // 0b111 -> two persons counting
        break;
        default             : ;
     }
    In_work = 0;                               // reset working variable
    TimerIn = 0;                               // reset timer
  }
}
```

Solve TPS signals with algorithm

For more than three TPS 210 devices following algorithm would be better than a table. The algorithm checks the devices in a row. If the current device is zero, and its neighbour device one a new person is detected. When the following device still one, the person has crossed two TPS devices.

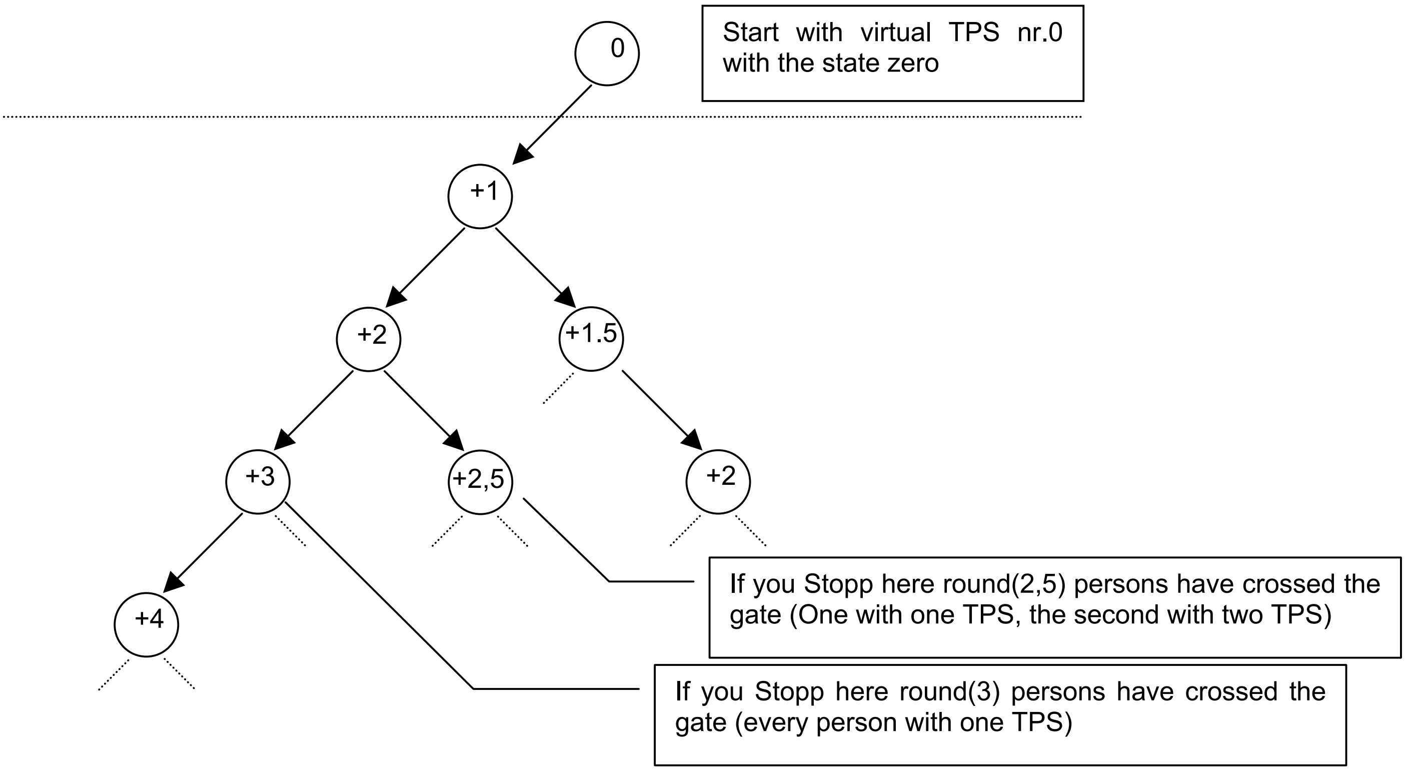
Check device by device in the row

TPS nr. 1	TPS nr. 2	TPS nr. 3	TPS nr. 4	TPS nr. 5	… …	TPS nr. x



go right if current device is one and next device is one

go left if current device is zero and next device is one



Start with virtual TPS nr.0 with the state zero

If you Stopp here round(2,5) persons have crossed the gate (One with one TPS, the second with two TPS)

If you Stopp here round(3) persons have crossed the gate (every person with one TPS)

Example for PC with Delphi:

```
...
if TimerIn > 250MS then                          // enters if 250 ms passed since last change
begin
  counter_In := 0;                                // reset counter var
  In_work := In_work shl 1;                      // shift left to solve problem with first mask values

  for j :=  0 to 6 do                            // check the 7 TPS devices
  begin
    mask := (In_work shr j) AND 3;                // mask input AND 00000011

    if mask = 2 then                             // that means a new person is detected
      begin
      counter_In := counter_In + 2;              // increment two -> for round(counter_In)
    end;

    if mask = 3 then                             // that means a person is detected with two devices
    begin
      counter_In := counter_In + 1;              // increment one -> for round(counter_In)
      end;
    end;

    counter_In := counter_In div 2;              // round(counter_In)
    result_In :=  result_In + counter_In;        // save counted people in result_In

    In_work    := 0;                             // reset working variable
    TimerIn    := 0;                             // reset
  end;
...
end;
```