ECE 8873 : Project 1
Differential Encoder
# Table of Content

*Table of Content* .......................................................................................................................... 2  
*Introduction* .................................................................................................................................. 3  
*Position Detector* .......................................................................................................................... 4  
*Electronic Circuit* .......................................................................................................................... 5  
*Part List* ......................................................................................................................................... 7  
*Algorithm* ...................................................................................................................................... 8
Introduction

There are several techniques to compute a mobile position and velocity. One of these techniques is to estimate the angular rotation of the wheels. From this information the relative movement of the mobile can be inferred and the position can be computed. The velocity is the computed depending on the movement of the mobile over a time period. The most common method used to estimate the angular position of the wheels consists in measuring the angular rotation thanks to an incremental encoder or absolute encoders. These devices are usually very accurate but they are too expensive. The solution presented in this report consists in using optical forks. Optical forks are widely used in computer mouse and are inexpensive. The information gathered by the captors will then be processed in an electronic circuit to make it readily available to the micro controller.

Using the information gathered by each of the wheel captors, the micro controller can deduce the relative position and velocity of the mobile using an appropriate algorithm.

This project was broken down into three main parts: the design of the captor, the design of the electronic circuitry and the algorithm.
Position Detector

The first step for this project is to design a captor capable of measuring the angular movement of a wheel. The solution of an incremental being too expensive an optical fork and a coded wheel can be used. The coded wheel is constituted of a succession of transparent and opaque streaks. A wheel needs to be fixed to each of the wheel of the mobile. The optical fork is an opto-electronic device composed of an IR emitter and receiver. The emitter and receiver are mounted head to head on a fork. When the wheel spins the coded wheel which is positioned between the emitter and receiver alternatively blocks the IR signal. Therefore the IR receiver generates an alternative signal relative to the angular rotation of the wheel.

The coded wheel can be printed on a transparent using a high resolution laser printer and then glued or fixed to the solid wheel. This solid wheel will be fixed to the mobile wheel.

Figure 1: example of a coded wheel

The optical fork chosen for the captor is a HOA 1874 from Honeywell and can be purchased at Radiospares for 4,29 €.

For the component to work at optimum settings resistors must be calculated accordingly depending on the power source available. The Talrik platform provides us with a 5V regulator. The emitter diode operates at a maximum of 50 mA of direct current. Therefore the resistor for the emitter must at least be of 100 Ω.

The maximum current for the receptor transistor is 30 mA, to reach such a current with a 5V Power supply we need a resistor of 166 Ω.

According to the technical specifications the lower the load resistance the lower the response time, to achieve good performance a low value resistance must be chosen. The following values will be used for the resistors: $R_1 = 110 \, \Omega$, $R_2 = 200 \, \Omega$. 
Electronic Circuit

To determine the angular rotation of the wheel a circuit must count the numbers of impulsion sent by the optical captor. A single impulsion directly corresponds to an angular measure depending on the resolution of the coding wheel. Most integrated circuits such as binary counters and latches require TTL signals or pulse signals. The signal generated by the optical fork resembles a sinus. Before sending the signal to a binary counter, it must be processed. Using an operational amplifier in voltage comparator mode the signal can be transformed into a rectangular pulse signal.

This pulse signal is then sent to a 4 bits binary counter, the carry line from the first binary counter is plugged into the CLK input of a second binary counter. This results in a global 8 bits counter.

The 8 output lines from the two binary counters are sent to a 3 states 8 bits latch. This latch acts as a memory buffer. The memory value in the circuit will be updated as soon as signals from the counters are received. The micro controller will then be able to read the current value of the latch using a polling method. To read the value of the latch the 8 outputs of the latch must be connected to the data bus available on the Data Bus header of the Talrik II. The memory circuit can directly be accessed using a predefined address. For instance to access IS1 the address required is FFB9. Some addresses ranges are left unused in order to extend the functionalities of the robot.

Below is the block diagram corresponding to the complete circuit:
The parts needed to build the circuit are the following:

<table>
<thead>
<tr>
<th>Designation</th>
<th>Part Reference</th>
<th>Constructor</th>
<th>Reseller</th>
<th>Quantity</th>
<th>Price (each)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 bits Binary Counters</td>
<td>74AC161PC</td>
<td>Fairchild Semiconductor</td>
<td>Radiospares</td>
<td>4</td>
<td>1,61€</td>
</tr>
<tr>
<td>8 bits Latch</td>
<td>74AC373PC</td>
<td>Fairchild Semiconductor</td>
<td>Radiospares</td>
<td>2</td>
<td>1,12 €</td>
</tr>
<tr>
<td>Operational Amplifier</td>
<td>LM358N</td>
<td>National Semiconductor</td>
<td>Radiospares</td>
<td>1</td>
<td>0,72 €</td>
</tr>
</tbody>
</table>

Below is the complete electronic circuit, with the various components value:

\[
\begin{align*}
R1 & = 110 \, \Omega \\
R2 & = 200 \, \Omega \\
R3 & = 10 \, k\Omega \\
R4 & = 10 \, k\Omega \\
RP & = 10 \, k\Omega 
\end{align*}
\]
# Part List

Below is the complete part list with the costs to build the encoder.

<table>
<thead>
<tr>
<th>Designation</th>
<th>Part Reference</th>
<th>Constructor</th>
<th>Reseller</th>
<th>Qty</th>
<th>Price (each)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optical Fork</td>
<td>HOA 1874</td>
<td>Honeywell</td>
<td>Radiospares</td>
<td>2</td>
<td>4.29 €</td>
</tr>
<tr>
<td>4 bits Binary Counters</td>
<td>74AC161PC</td>
<td>Fairchild Semiconductor</td>
<td>Radiospares</td>
<td>4</td>
<td>1.61 €</td>
</tr>
<tr>
<td>8 bits Latch</td>
<td>74AC373PC</td>
<td>Fairchild Semiconductor</td>
<td>Radiospares</td>
<td>2</td>
<td>1.12 €</td>
</tr>
<tr>
<td>Operational Amplifier</td>
<td>LM358N</td>
<td>National Semiconductor</td>
<td>Radiospares</td>
<td>1</td>
<td>0.72 €</td>
</tr>
</tbody>
</table>

The price of the various components needed to complete the circuit is not included here as they are not significant. (i.e.: price of the resistors)
Algorithm

Let \( dw \) be the distance between the wheel.
Let \( R \) the radius of one wheel (in the case of the Talrik II robot we have \( R = 1.5 \) inches)
Let \( N \) be the number of impulsions corresponding to one full rotation of the wheel (\( N = \) Resolution \( \times \) 360)

Let \( x \) and \( y \) be the current coordinates of the mobile, and \( \theta \) the angle between the direction of the robot and the \( x \) axis.
Let \( xr \) and \( yr \) be the coordinates of the right wheel.
Let \( xl \), and \( yl \) be the coordinates of the left wheel.

For this algorithm we will consider that when the robot wants to turn right his left wheel will be blocked and no impulsions will be recorded for this wheel.
In the same manner for a left turn we will consider the right wheel to be “inactive”.
We will use a Timer Class in order to estimate the time elapsed between successive calls to the timer function.
This timer will be used to compute the average speed of the mobile.

This will lead to 3 possibilities:
Neither wheel is blocked: the robot is on a straight path.
The Left wheel is blocked: the robot does a right turn.
The right wheel is blocked: the robot does a left turn.

\[
\text{Const double } R = 1.5 * 2.5 ; \\
\text{Const double } dw = ???; \\
\text{Const Int } N = ???; \\
\text{Float dist}; \\
\text{Float velocity}; \\
\text{Int n_measured} \\
\text{CTimer m_oRobotTimer;}
\]

If (Wheel.Right.Blocked == 0 && Wheel.Left.Blocked == 0)
{
    n_measured = Captor.Wheel.Right.Read() + Captor.Wheel.Left.Read();
    n_measured /= 2;
    Dist = 2 * Pi * R * n_measured / N;
    x += Dist * \cos(\theta);
    y += Dist * \sin(\theta);
    velocity = Dist / m_oRobotTimer.Elapsed();
}
If (Wheel.Right.Blocked == 1 && Wheel.Left.Blocked == 0)
{
    n_measured = CaptorCaptor.Wheel.Left.Read();
    xl = x – (dw /2) * cos (90 - θ);
    yl = y + (dw /2) * sin (90 - θ);
    θ += 360 * R * n_measured / (N * dw);
    x = (x – xl) * cos (θ) - (y – yl) * sin (θ);
    y = (x – xl) * sin (θ) + (y – yl) * cos (θ);
}

If (Wheel.Right.Blocked == 0 && Wheel.Left.Blocked == 1)
{
    n_measured = CaptorCaptor.Wheel.Right.Read();
    xr = x + (dw /2) * cos (90 - θ);
    yr = y - (dw /2) * sin (90 - θ);
    θ -= 360 * R * n_measured / (N * dw);
    x = (x – xr) * cos (θ) - (y – yr) * sin (θ);
    y = (x – xr) * sin (θ) + (y – yr) * cos (θ);
}