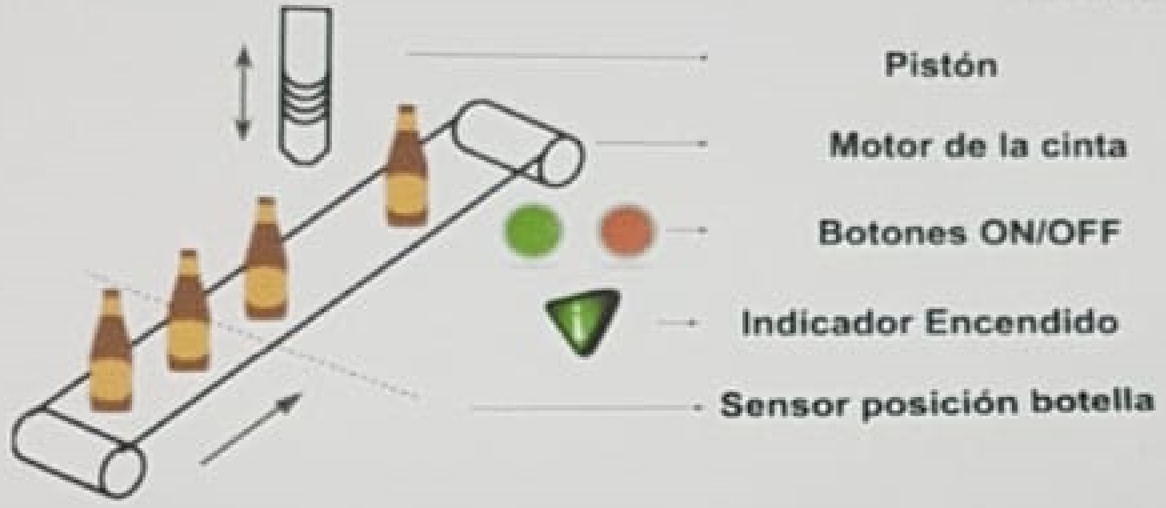
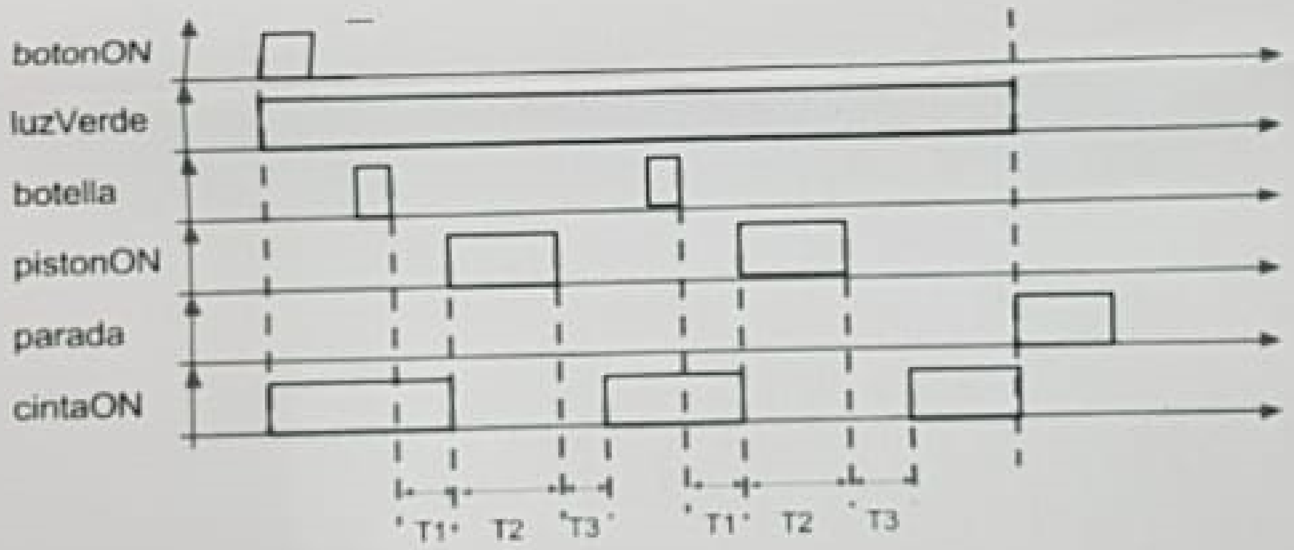
FINAL INFORMATICA II	Total Hojas	Duración	23/May/19
		19:00 a 21:00 hs	

Nombre y Apellido	Nº Legajo	Calificación	Docente evaluador	
			Nombre	Firma
[redacted]	[redacted]	10 (diez)	Trujillo	

Se diseñó un equipo para realizar el tapado a presión de botellas de vidrio. El mismo se basa en una cinta transportadora que lleva las botellas hasta un pistón neumático, que luego baja colocando la tapa de la botella a presión, y finalmente sube para permitir que la cinta siga moviéndose y vuelva a repetirse la operación. El sistema cuenta además con un botón de encendido y uno de parada. El diagrama del sistema y el funcionamiento de sus entradas y salidas viene dado por los siguientes esquemas:





Esto es, al detectarse la opresión del botón de encendido, se enciende el un indicador y se pone en marcha la cinta. Esta continuará moviéndose hasta que el sensor de posición encuentre una botella. Una vez que suceda esto, se deberá esperar un tiempo T1 hasta que la botella se encuentre justo debajo del pistón. Pasado este tiempo se detendrá la cinta, y se accionará el pistón neumático durante un tiempo T2, tiempo que se considera suficiente para que el pistón ejerza la fuerza suficiente para taponar la botella. Finalmente se desactivará el pistón y se dará un tiempo T3 para que el mismo libere el pico de la botella antes de volver a accionar la cinta y comenzar el proceso nuevamente. El sistema permanecerá relizando esta operatoria hasta detectar el botón de parada, momento en el cual se detendrán todas las actuaciones.

Para hacer el sistema independiente de sus componentes mecánicos, los tiempos T1, T2 y T3 podrán configurarse desde una interfaz gráfica en una PC, conectada al sistema embebido de control por medio de su puerto serie, configurado en 9600,8,N,1. El formato de la trama de configuración viene dado por el siguiente esquema:

'$'	T1 (msegs) (2bytes)	T2 (msegs) (2bytes)	T3 (msegs) (2bytes)	'#'

[redacted]

[redacted]

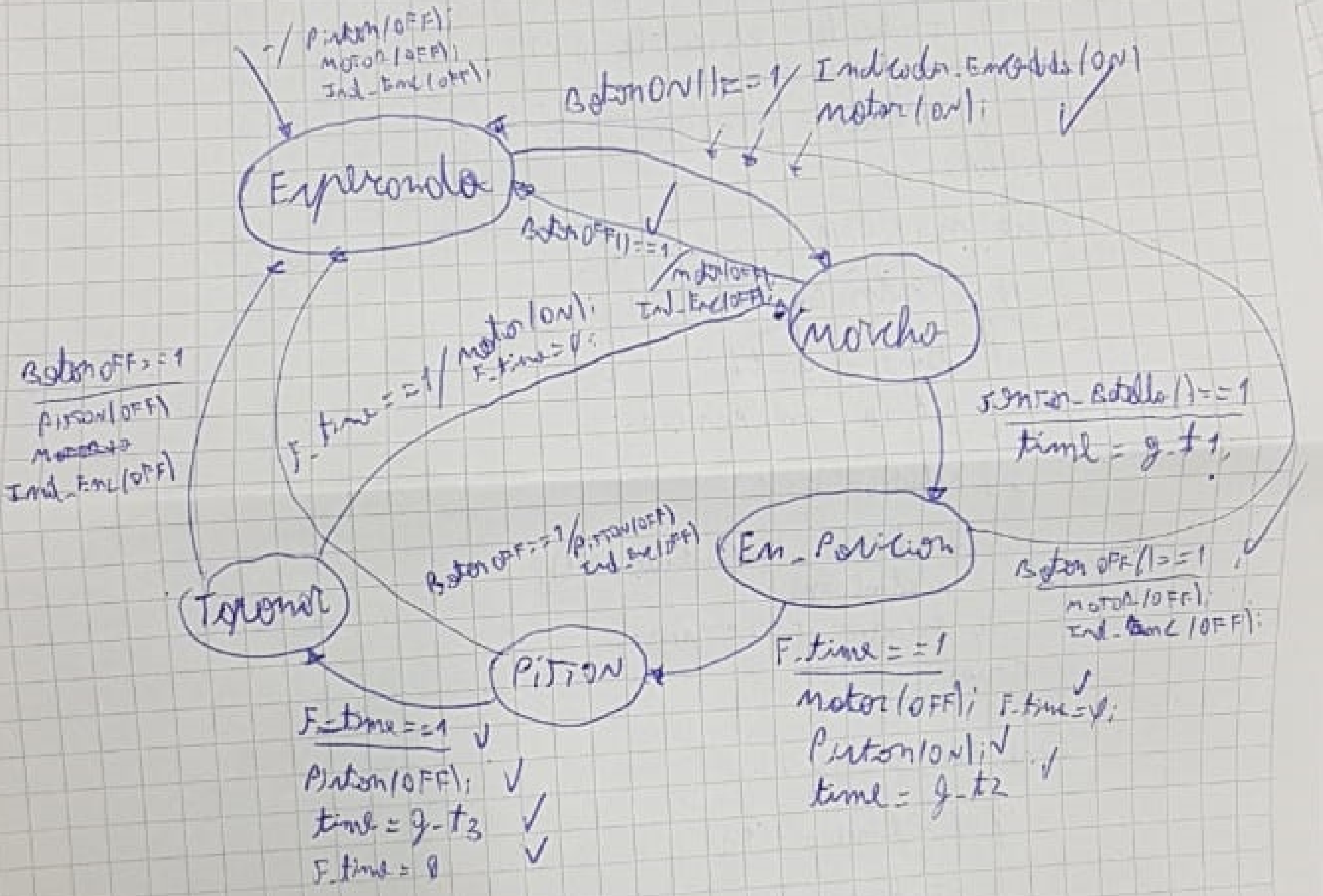
1)

Entradas:

- Boton_ON()
- Boton_OFF()
- sensorBotella()

Salidas

- Piston()
- motor()
- Indicador_Encendido()



```
void maq_estado()
{ static uint8_t estado = ESPERANDO;

  switch (estado)
  {
      case ESPERANDO:
          if (BotonON()==1)
          { Indicador_Encendido(ON);
            motor(ON);
            estado = MARCHA;
          }
      break;

      case MARCHA:
          if (sensor_Botella()==1)
           { time = g_t1;
             estado = EN_POSICION;
           }
          if (BotonOFF()=1)
           { Indicador_Encendido(OFF);
             motor(OFF);
           } estado = ESPERANDO;
      break

      case EN_POSICION:
          if (F_time == 1)
          {    F_time == 0;
               motor(OFF);
               Piston(ON;
           }   time = g_t2;
          }     estado = PISTON;
          if (BotonOFF()==1)
            { Indicador_Encendido(OFF);
              motor(OFF)
              estado = ESPERANDO;
            }
      break;

        }
```

[redacted]

```
case PISTON:
   if( F.time==1)
    { F_time=0;
      Piston(OFF);
      time= g_t3;
      estado=TAPONAR;
    }
   if ( botonOFF()==1)
    { piston(OFF);
      Indicador_Encendido(OFF);
      estado= ESPERANDO;
    }
   break;

 case TAPONAR;
   if (F_time ==1 )
    { motor(ON);
      F_time=0;
      estado = MARCHA;
    }
   if ( BotonOFF()==1)
    { Piston(OFF);
      Indicador_Encendido(OFF);
      estado= ESPERANDO
    }
   break;

  Default:
      piston(OFF);
      motor(OFF);
      Indicador_Encendido(OFF);
 }
}

uint8_t  BotonON()                                   void  Piston(estado)
 {  returng getPin(BOTONON);                          { setPin(PISTON, estado);
                                                       }
 }
uint8_t BotonOFF()
 { return getPin(BOTONOFF);
 }
```

NOTA

2)

```
void Systick_IRQHandler ()
{
     time--;
     if (time == 0)
       { F_time = 1;
       }
 }
```

3)

```
void Maquina_RX()
{
     static uint8_t estado = HEADER;   uint16_t t1, t2, t3;
     uint16_t dato;

     if ((dato = PopRx()) != 0)
     { switch (estado)
        {
           case HEADER:
             if (dato == 't')
                estado = T1;
           break;

           case T1_1:
               t1 = dato
               estado = T1_2;
             break;
           case T1_2:

               t1 = (uint16_t) (t1 << 8) | dato;
               estado = T2_1;
             break;
          case T2_1;
                t2 = dato
                estado = T2_2;
              break;
```

[redacted]

```
        Case T2_2:
            t2 = (int16_t) (t2 << 8) | dato;
            estado= T3_1;
            Break;

         Case T3-1:
               t3 = dato
               estado= T3_2;
              Brea;
         Case T3_2:

               t3 = (int16_t) (t3 << 8) | dato;
               estado= FIN;
              Break;

          Case FIN:
              if (dato == '#')
              {    g_t1 = t1;
                    g_t2 = t2;
               }   g_t3 = t3;
               estado= HEADER;

              Break;
           default:
        }
     }
}

  int  main()
  {   while(1)
       {
           maq_Emisor();
           Maquina RX()
       }
  }
```

NOTA

[redacted]

[redacted]

2)

```
void UART0_IRQHandler ()
{
        uint8_t iir, dato;

    do
     {
            if (iir == 0x02)
            {   THR0RX = 0;
                dato = popRX
            }
            else
                  g_RX_Alort = 0;
            if (iir == 0x04)
              {
                  ~~[illegible]~~
                  pushRX (&dato)
              }   THR0TX = 0;
     }
     while ( iir != 0x04);
}
```