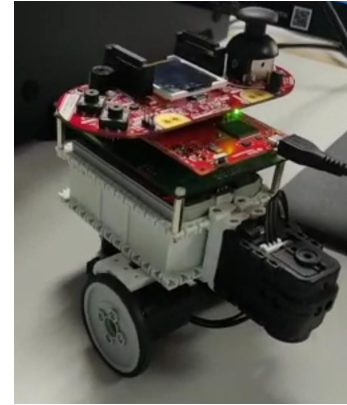


PROGRAMACIÓ D'ARQUITECTURES ENCASTADES



GRUP 09: Maria Pérez Pullès, Oscar de Caralt Roy, Maria Ruiz Àvila

Índex

- Introducció
- Estructura del codi
- Funcionalitats
- Problemes sorgits
- Conclusions

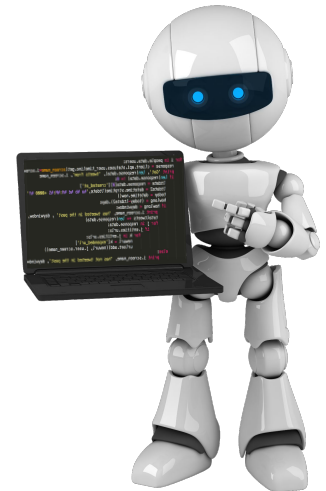


Introducció

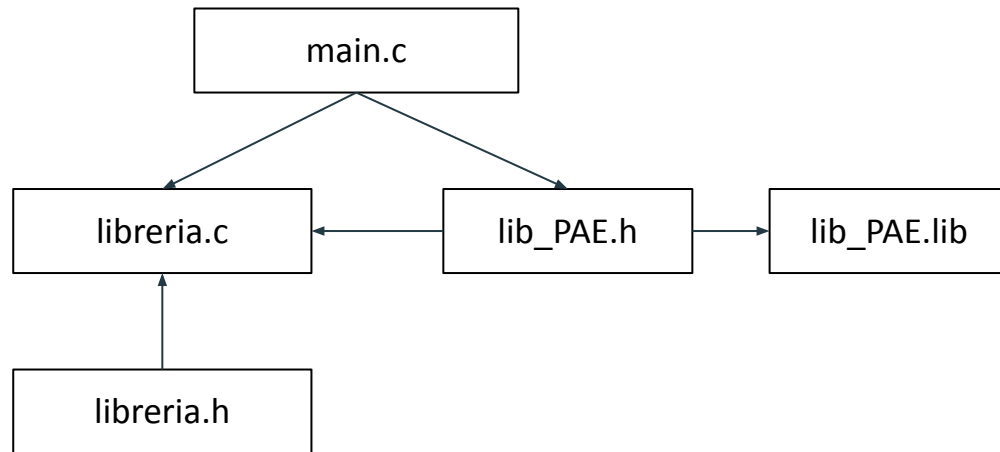
Evolució de la P3

Per a implementar el moviment autònom en l'habitació.

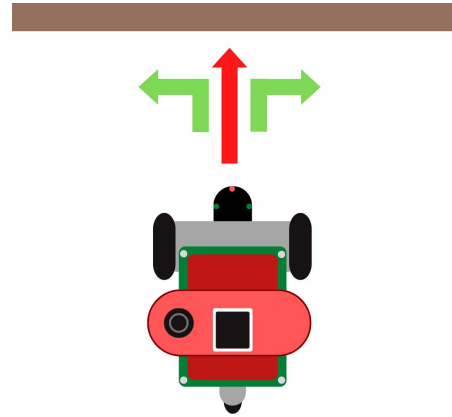
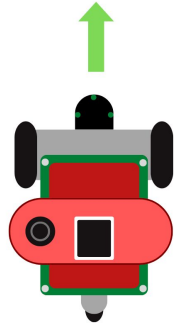
- El robot haurà d'anar a buscar la paret més propera.
- Un cop trobada, l'ha d'anar resseguint sense en cap cas col·lidir amb ella. Ha de mantenir sempre a una distància de seguretat raonable respecte la paret.
- Haurà de superar les diferents corbes possibles que es pot trobar en les cantonades.
- En cap cas haurà de col·lidir amb els obstacles.



Estructura del codi

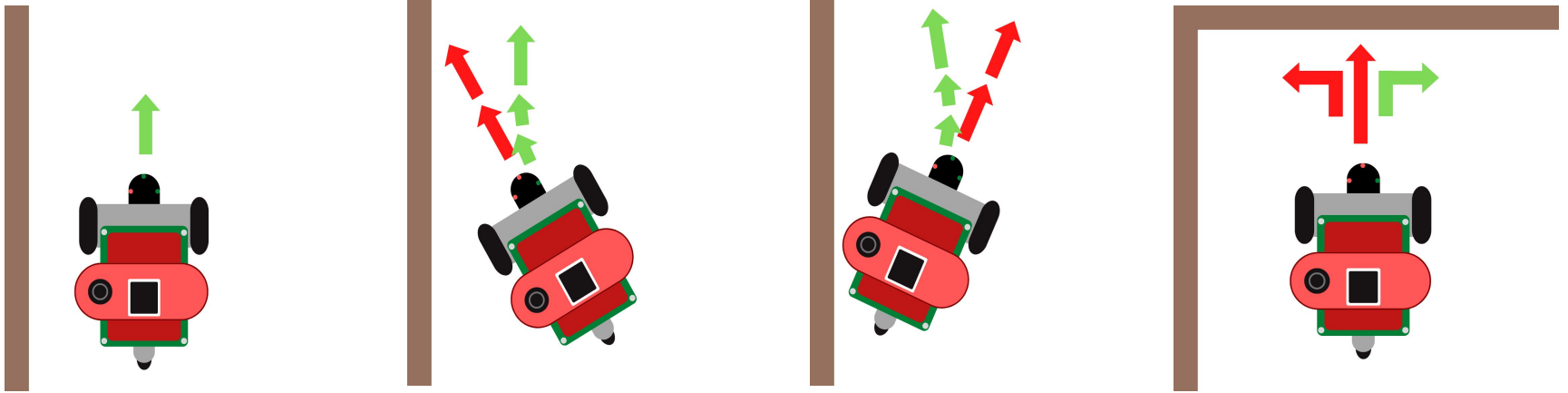


Funcionalitats: Ruta amb algorisme implementat



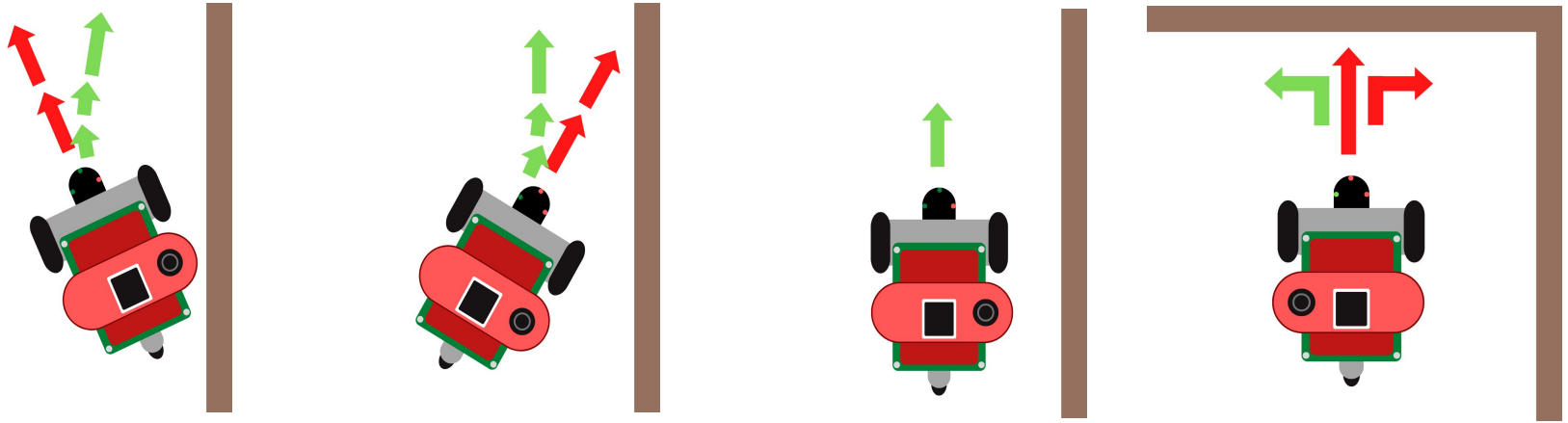
Funcionalitats: Ruta amb algorisme implementat

Case PARED == esquerra



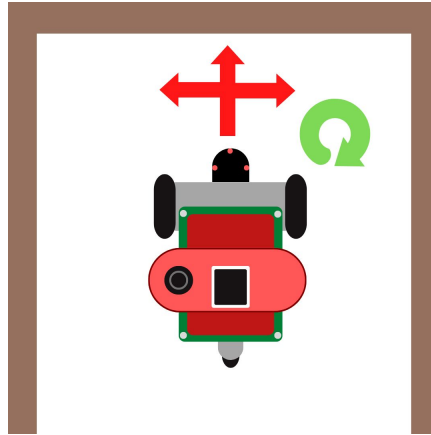
Funcionalitats: Ruta amb algorisme implementat

Case PARED == dreta



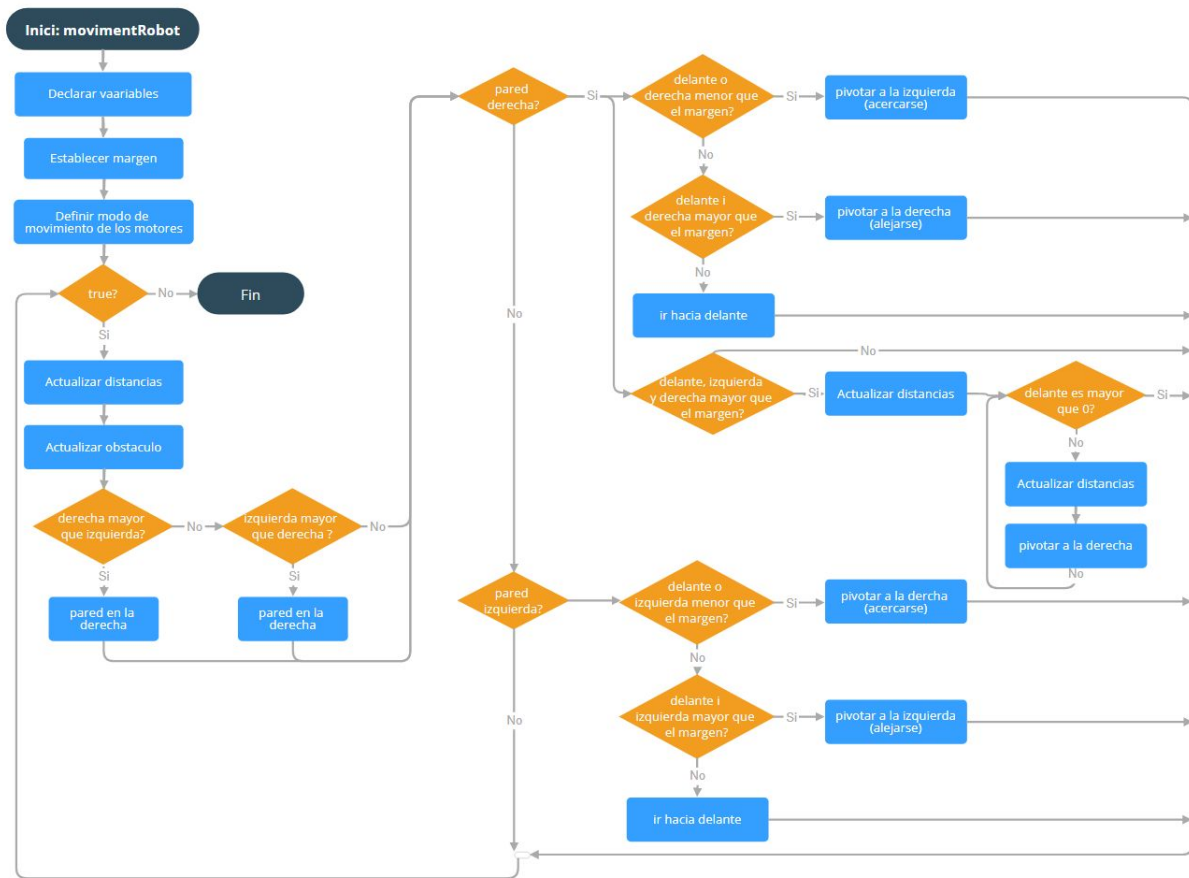
Funcionalitats: Ruta amb algorisme implementat

Case PARED == dreta



Funcionalitats: Ruta amb algorisme implementat

```
void recorrido(void) {  
    PARED pared;  
    byte obstaculo;  
    byte del, derecha, izquierda;  
    byte margen = 15;  
    setEndlessTurn();  
  
    while (1) {  
        distanciaGeneral(&del, &derecha, &izquierda);  
        obstaculoDetectado(&obstaculo);  
  
        if (derecha > izquierda) {  
            pared = DERECHA;  
        } else if (izquierda > derecha) {  
            pared = IZQUIERDA;  
        }  
  
        switch (pared) {  
            case DERECHA:  
                if (del > margen || derecha > margen) {  
                    pivotarIzquierda();  
                } else if (del < margen && derecha < margen) {  
                    pivotarDerecha();  
                } else {  
                    delanteV(100);  
                }  
                if (del > margen && izquierda > margen && derecha > margen) {  
                    distanciaGeneral(&del, &derecha, &izquierda);  
                    while (del > 0) {  
                        distanciaGeneral(&del, &derecha, &izquierda);  
                        pivotarDerecha();  
                    }  
                }  
                break;  
            case IZQUIERDA:  
                if (del > margen || izquierda > margen) {  
                    pivotarDerecha();  
                } else if (del < margen && izquierda < margen) {  
                    pivotarIzquierda();  
                } else {  
                    delanteV(100);  
                }  
                break;  
        }  
    }  
}
```



Funcionalitats: Comunicació

```
byte TxPacket (byte bID, byte bParameterLength, byte bInstruction, byte Parametros[16]);  
void TxUAC0(byte bTxdata);  
void TxUAC2(byte bTxdata);  
struct RxReturn RxPacket(void);  
struct RxReturn RxTxPacket(byte bID, byte bParameterLength, byte bInstruction, byte Parametros[16]);
```

Funcionalitats: declaracions i gestió d'interrupcions

```
void Activa_TimerA0_TimeOut(void);  
void Reset_Timeout(void);  
void Stop_Timeout(void);  
byte TimeOut(uint16_t t);  
void EUSCIA0_IRQHandler(void);  
void EUSCIA2_IRQHandler(void);  
void init_uart(void);  
void init_interrupciones(void);  
void init_timers(void);  
void init_GPIOs(void);
```

Funcionalitats: LEDS

```
//LEDS  
void encender_LED_derecho(void);  
void encender_LED_izquierdo(void);  
void encender_LEDS(void);  
void apagar_LED_derecho(void);  
void apagar_LED_izquierdo(void);  
void apagar_LEDS(void);  
void leer_LED_derecho(byte *readVal);  
void leer_LED_izquierdo(byte *readVal);
```

Funcionalitats: moviment

```
//movimiento
void setEndlessTurn(void);
void delante(void);
void atras(void);
void delanteV(byte velocidad);
void atrasV(byte velociad);
void parar(void);

//giro
void giroDerecha(void);
void giroIzquierda(void);
void pivotarDerecha(void);
void pivotarIzquierda(void);

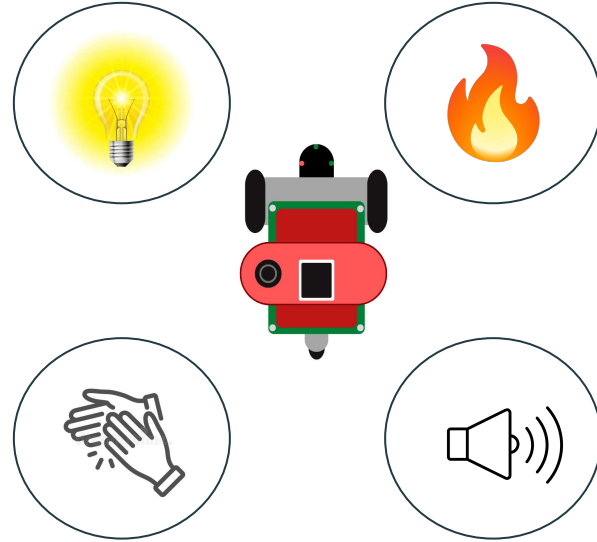
//sensores de distancia
void distanciaDelante(byte *readVal);
void distanciaDerecha(byte *readVal);
void distanciaIzquierda(byte *readVal);
void distanciaGeneral(byte *delante, byte *derecha, byte *izquierda);
```

```
void escribir(char mensaje[], uint8_t linea);
void borrar(uint8_t linea);
void melodia();
void sonido(byte *readVal);
void obstaculoDetectado(byte *readVal);
void luzIzquierda(byte *readVal);
void luzDelante(byte *readVal);
void luzDerecha(byte *readVal);
void leerVelocidad(uint16_t *derecho, uint16_t *izquierdo);
void temperatura(byte *readVal);
void luzDetectada(byte *readVal);
void silencio();
void sonidoConstante(void);

//void girarGradosDerecha (byte grados);
//void girarGradosIzquierda (byte grados);
void pivotarDerechaV (byte velocidad);
void pivotarIzquierdaV (byte velocidad);
void girarDerechaV (byte velocidad);
void girarIzquierdaV (byte velocidad);
```

Funcionalitats extres

- Joystick per interactuar amb el menú
- Recorregut de llum
- Detector de temperatura
- Mostrar velocitats del motor
- Arrancar amb palmada (finalment inhabilitada)
- Melodia



Problemes que han sorgit

- Problemes amb el robot assignat (4)
- Refer el codi practicament desde 0
- Temps



Conclusions

