```
// Explanation of functions


// Table of contents:
// OUTPUTS:
   dispNum(x,y, value);
   dispText(x,y,"example text" );
   PlayTone(frequency, duration);

// TIMING & RANDOMNESS
   startTimer1();
   readTimer1();
   wait(time);
   random(minimum, maximum);

// SENSORS
   reflectionDown();
   reflectionRedLeft();
   ultrasound();
   blink        // is an int variable

// MOVEMENT
   motor(speed_left_belt, speed_right_belt);
   turn(speed, degrees);


// EXAMPLE SYNTAX CODE
```

```
// Output for FEEDBACK

void dispNum(int x, int y, int value)
{
 // displays a numerical value on the screen of the robot at the
 // screen coordinates x (horizontal) and y (vertical).
 // In x the display is 90 pixels wide and in y it has 8 lines that can be adressed
 // with LCD_LINE1, LCD_LINE2, LCD_LINE3, ..., LCD_LINE8.
}
 //example:
 dispNum(0,LCD_LINE1, 42);



void dispText(int x, int y, "example text" )
{
 // displays text on the screen of the robot at the
 // screen coordinates x (horizontal) and y (vertical).
 // In x the display is 90 pixels wide and in y it has 8 lines that can be adressed
 // with LCD_LINE1, LCD_LINE2, LCD_LINE3, ..., LCD_LINE8.
}
 //example:
 dispText(0,LCD_LINE1, "Hello World" );



void PlayTone(int frequency, int duration)
{
 // plays a tone with the frequency specified in variable "frequency" in Hz
 // for the time specified in "duration" in ms.
}
 //example:
 PlayTone(440,200);
```

```
// TIMING and RANDOMNESS

void startTimer1()
{
 // starts or restarts a timer when this function is called.
 // start_timer2() and start_timer3() work analogously.
}
 //example:
 startTimer1();



unsigned long int readTimer1()
{
 // returns the time in milliseconds that has passed after starting the timer.
 // readTimer2() and readTimer3() work analogously.
}
 //example:
 int time = readTimer1();



void wait(int time)
{
 // pauses the execution of the program for the time specified
 // in the parameter "time" in milliseconds.
}
 //example:
 wait(1000);



int random(int minimum, int maximum)
{
 // returns a random number between minimum and maximum
 // if minimum is greater than maximum, it returns 0
}
 //example:
 int x = random(-20,20);
```

```
// SENSORS

int reflectionDown()
{
 // This sensor emits red light downwards and measures how much is reflected
 // It returns values between 0 and 100, where higher numbers mean higher reflectivity
 // of the surface below
}
 //example:
 int refl = reflectionDown();



int reflectionRedLeft()
{
 // The color reflection sensors send out red light and detect how much of
 // the emitted red light is reflected back into the sensor.
 // Red objects reflect more red light than green or blue objects.
 // Smaller values correspond to less reflectivity of red light.
 // This function return a reflection value for the left reflection sensor.
 // A similar function exists for the right reflection sensor and is called by
 // reflectionRedRight()
}
 //example:
 int reflRight = reflectionRedRight();



int ultrasound()
{
 // The ultrasound sensor measures the distance between the sensor and an object
 // that reflects ultrasound by detecting the time of flight of an ultrasound pulse.
 // It returns a number that correlates with the distance in cm.
}
 //example:
 int ultra = ultrasound();


int blink;
// The reflection sensors in the front can also sense a blinking light source.
// This process runs continuously in the background.
// The average value from the last one and a half seconds is stored in the
// global integer variable "blink" that can be used throughout the code.
// example:
   if(blink > 2)
```

```
// MOVEMENT


void motor(int speed_left_wheel, int speed_right_wheel)
{
 // the speeds can be values between -100 (full backwards) and 100 (full forward).
 // Speed of 0 stops the motor.
}
 // example:
 motor(10,-10);



void turn(int speed, int degree)
{
 // Sets one belt to positive speed and the other to negative speed and thereby
 // turns the robot with the motor speed defined in the first parameter.
 // The turning direction and stopping angle is defined in degrees in the
 // variable degree. A positive degree value turns the robot clockwise.
}
 // example:
 turn(20,-90);
```

```
// EXAMPLE OF SYNTAX

int x = 7;
int y = 42;

// decleration of a function
int function(int parameter1, int parameter2)
{
 return parameter1 + parameter2;
}


while(true)  // infinite loop
{
 if(x == y || x != 7)     // "==" equal, "||" logical or, "!=" not equal
 {
  x = 42;
 }
 else
 {
  y = 7;
 }

 while(x >= y && x > 7) // "<=" less than, ">=" greater than, "&&" logical and
 {
  x = x - 1;              // you could also write "x--;"
 }
}

/* this is
a multi-line
comment */
```