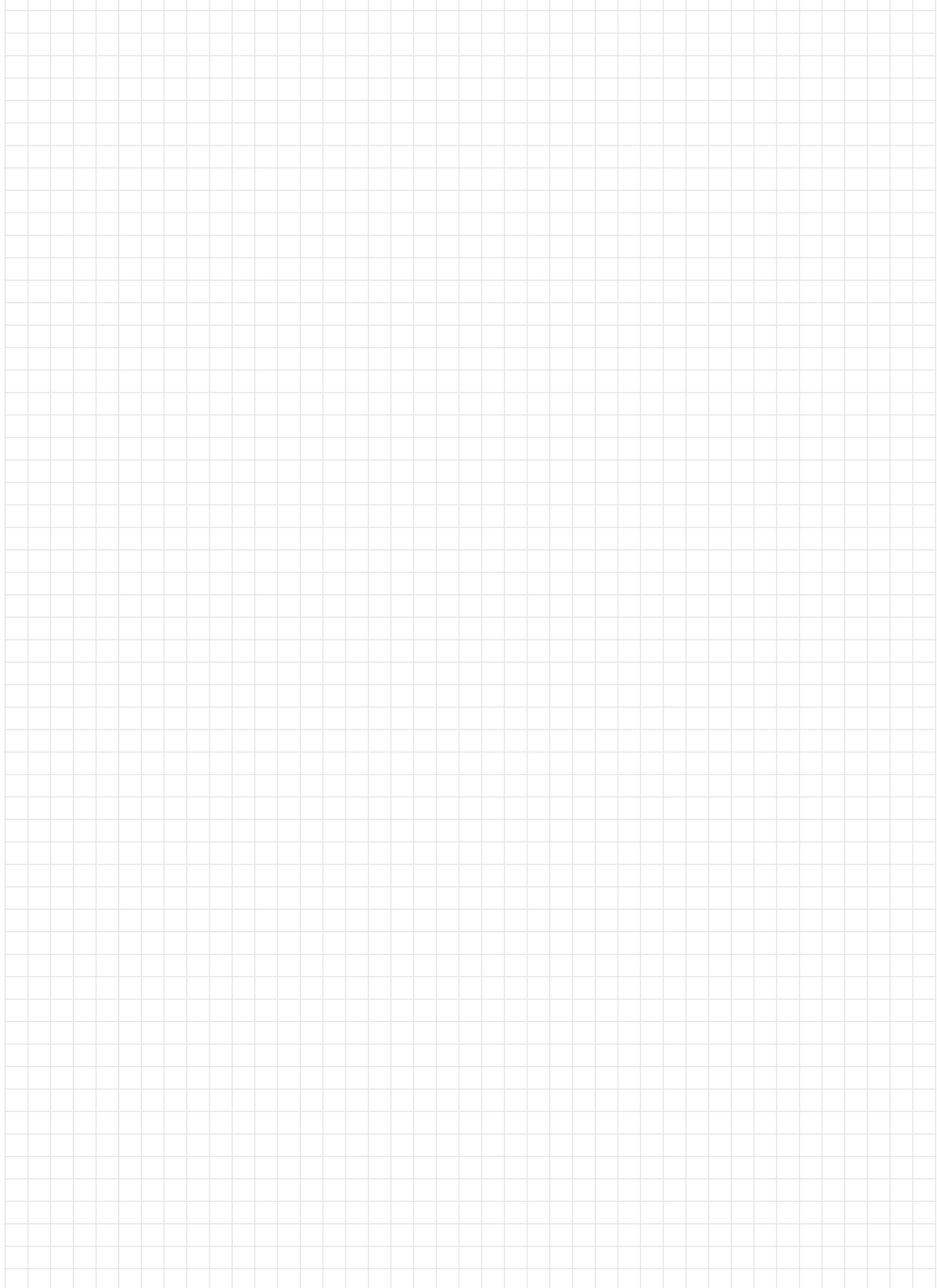


P U T P U T

Reclaim your urban environment playfully.



PUPUPUPUP



Urban II linigolf Device

Project Sheet

- | | |
|--------------------------------|----------------|
| • Konzept / Concept | Seiten 03 – 07 |
| • Recherche / Research | Seiten 08 – 09 |
| • Umsetzung / Realization | Seiten 10 – 15 |
| • Programmierung / Programming | Seiten 16 – 17 |

• Konzept

Das Thema Stadtraum und die Nutzung dessen sind immer wieder Gegenstand von Diskussionen und Aktionen der Politik, Architektur, Bürgerinitiativen aber auch im Bereich Gestaltung und Design steht das Thema Urbaner Raum seit Jahren im Fokus der Aufmerksamkeit.

Jeder Quadratmeter öffentlicher Raum ist stark umkämpft. Es wird geworben, ausgestellt, verkauft, gebaut, zerstört und renaturiert. Jeder möchte ein Stück dieser scheinbaren Kostbarkeit für sich und seine Zwecke erobern. Wie dies geschieht ist zwar nicht immer für alle Seiten zufriedenstellend soll aber in diesem Fall einmal nicht Gegenstand des Projektes sein.

Trotzdem spielt die Rückeroberung und Nutzung des Stadtraumes eine zentrale Rolle im Rahmen meines Projektes zum Thema „Game or Play“ während des Wintersemesters 06/07 an der UdK Berlin. Ziel des Semesterthemas war es interaktive Spielobjekte zu erschaffen.

Reclaim your urban environment playfully.

PUTPUT

PUTPUT ist nicht mehr und nicht weniger als ein mobiles Minigolfsystem.

Der Stadtraum bietet eine Vielzahl an natürlichen und künstlichen Hindernissen und ist so gesehen ein unerschöpflicher Minigolfparcour. Das hat vor einigen Jahren bereits eine junge Generation von Golfspielern für sich entdeckt. Das StreetGolfing oder UrbanGolfing kam in einigen Metropolen auf, verschwand aber auch relativ schnell wieder von der öffentliche Bildfläche.

PUTPUT setzt genau an dieser Stelle ein. Da Minigolf, eine doch weit verbreitete Unterart des Golfens, für jederman, egal ob sportlich versiert oder nicht, sich immernoch großer Beliebtheit erfreut, bietet es die perfekte Basis um auf spielerische und sich nicht allzuernstnehmende Art und Weise die Rückeroberung des öffentlichen Stadtraumes zu betreiben.

Das simple System von Minigolf ist für jederman klar verständlich und bedarf keiner langwierigen Trainingsphase Man benötigt lediglich einen Schläger, einen Ball und ein Ziel in das der Ball mit Hilfe des Schlägers eingelocht wird. Der große Reiz am Minigolf liegt in der Einfachheit des Spiels, der Verspieltheit des Parcours und natürlich an der sehr geselligen Komponente des Wettstreites unter Freunden. All das findet man auf Minigolfanlagen aber auch genauso gut auf Straßen, in Parkanlagen und anderen öffentlichen Plätzen. Und dazu noch kostenlos.

Das Ergebnis dieser Überlegungen ist eine sich autark bewegende Minigolf einheit, die auf äußere Gegebenheiten reagieren und erkennen kann ob ein Ball sein Ziel gefunden hat.

PutPut. Urban Minigolf Device.

• Concept

The subject of urban space and its utilisation does not only consistently concern discussions and campaigns in politics, architecture and citizen initiatives. For many years now urban space has been a subject of vital importance to the field of creative design as well.

Every single squaremetre of public space is heavily competed for. People advertise, exhibit, sell, construct, destruct and re-nature. Everybody wants to conquer a piece of this seemingly treasure for himself and his own purposes. The ways in which these recaptures run may not always be satisfactory to all sides, however, this will not be the subject of this project.

Still, the recapture and utilisation of urban environment plays a role of vital importance within the scope of my project on the subject of „game or play“ during winter semester 06/07 at the UdK Berlin. The aim of this semester's subject was to create interactive play objects.

Reclaim your urban environment playfully.

PUTPUT

PutPut is in effect a mobile miniature golf system.

Urban space features a variety of natural and artificial obstacles and thus serves as an inexhaustable miniature golf course.

A few years ago a young generation of golf players discovered the streets for their purposes. Street Golfing or Urban Golfing emerged in some large cities, but soon vanished into thin air again.

This is exactly where PUTPUT comes in. Miniature golf is still a rather widespread subspecies of golf, which enjoys great popularity among both sportive or less so people. It thus forms the perfect basis for reclaiming public urban space in a playful and not over-serious way.

The simple system of miniature golf is plain to everybody and goes without any tedious training. All one needs is a club, a ball and a target to pocket the ball. The great charm of miniature golf lies in its simplicity, in the playfulness of the course and, naturally, in the sociable quality of competing with friends. All this one may find on miniature golf courses, but just as well in the streets, in parks and in public spaces. And, what is more, it is then free of charge.

The result is a self-sufficiently moving miniature golf device, which is able to react to external conditions and which may detect if a ball has reached its target.

PUTPUT. Urban Minigolf Device.







1

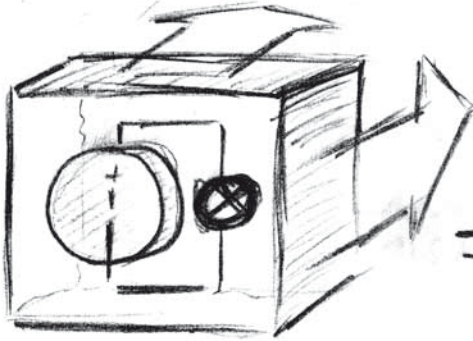


2

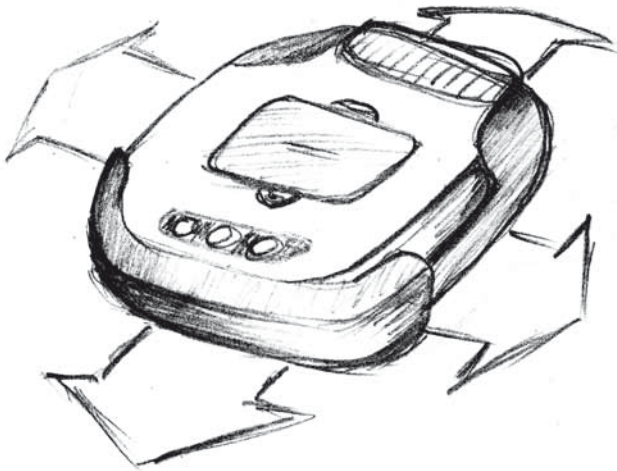
1 Markierungslampen zur Ausleuchtung und Abgrenzung des Parcours. Eingebauten Magnete ermöglichen die LED Boxen an metallische Untergründe zu heften. *LightBoxes with build in magnets for mettalic surfaces.*

2 Komplettsset bestehend aus 3 Schlägern, 3 beleuchteten Bällen, 4 Markierungsleuchten und dem «PUTPUT». *Set contains 3 putter, 3 lighted golfballs, 4 ledboxes and «PUTPUT».*

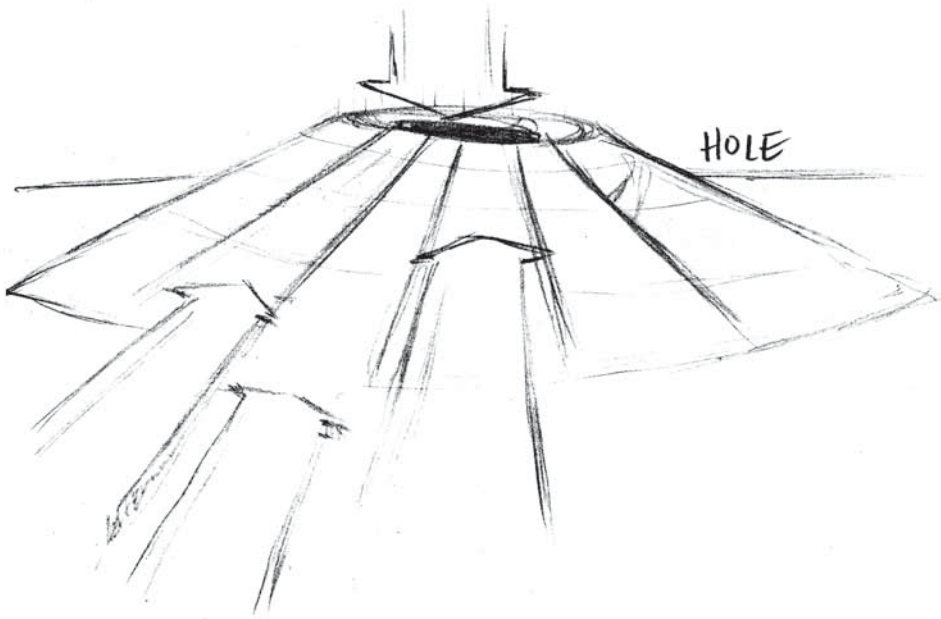
• Recherche / Research



LED-BOXES
To mark the playground



ROBOT
To move the Hole

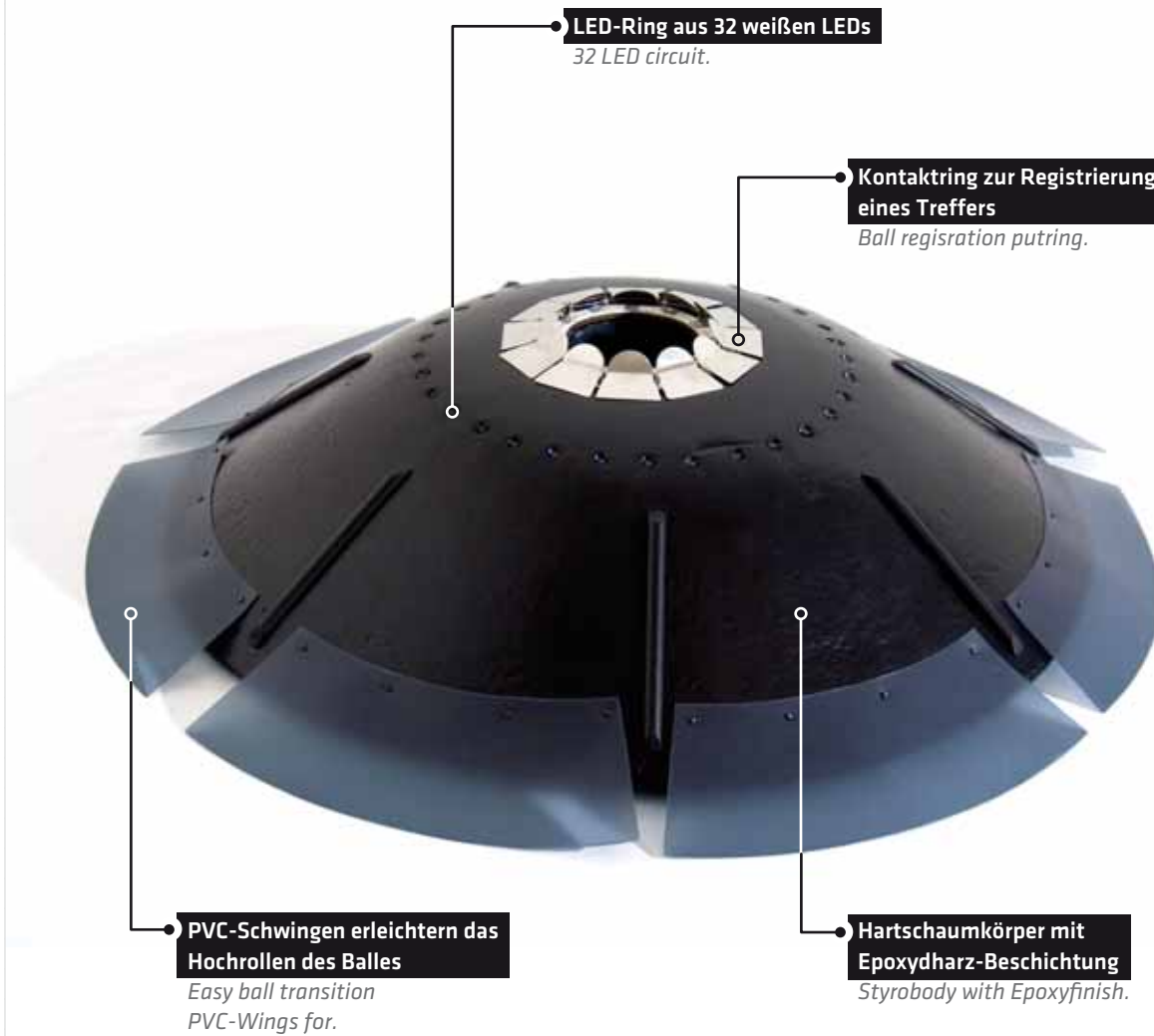


HOLE

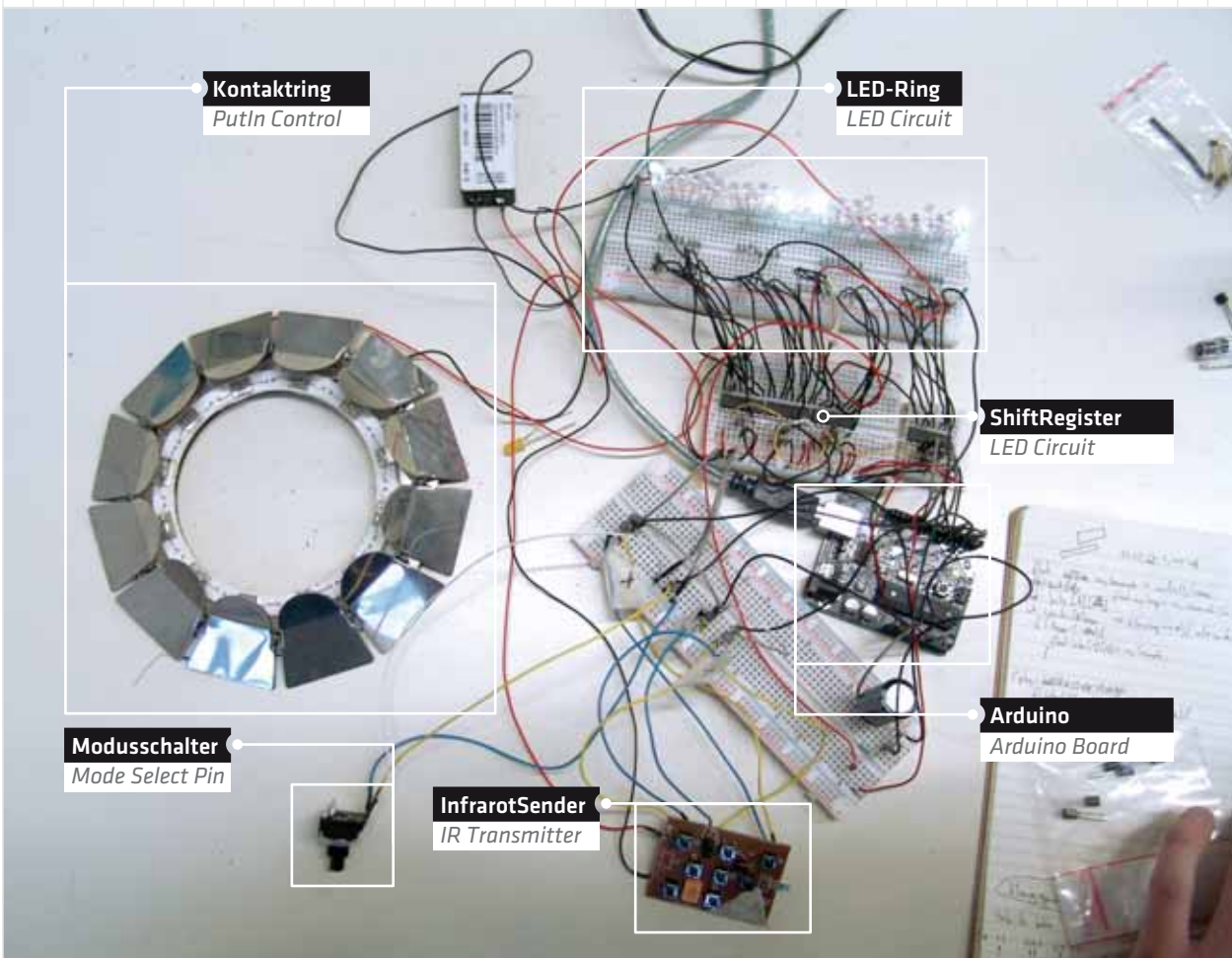
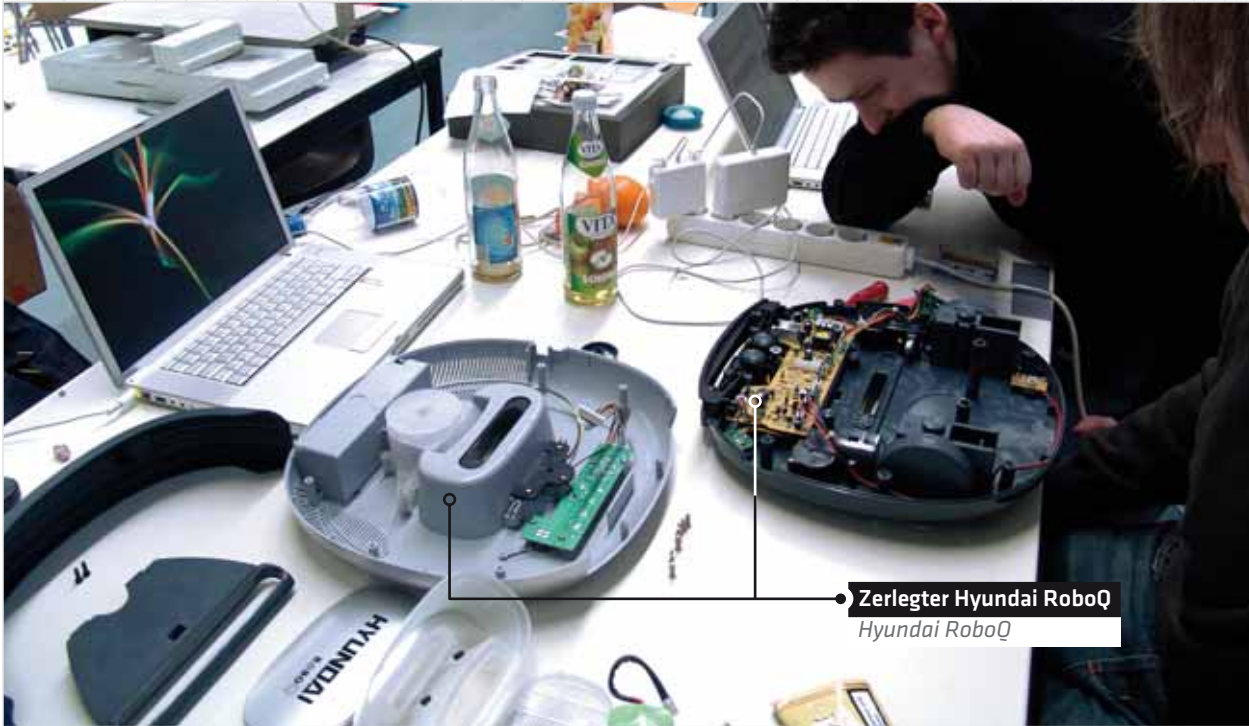
• Recherche / Research



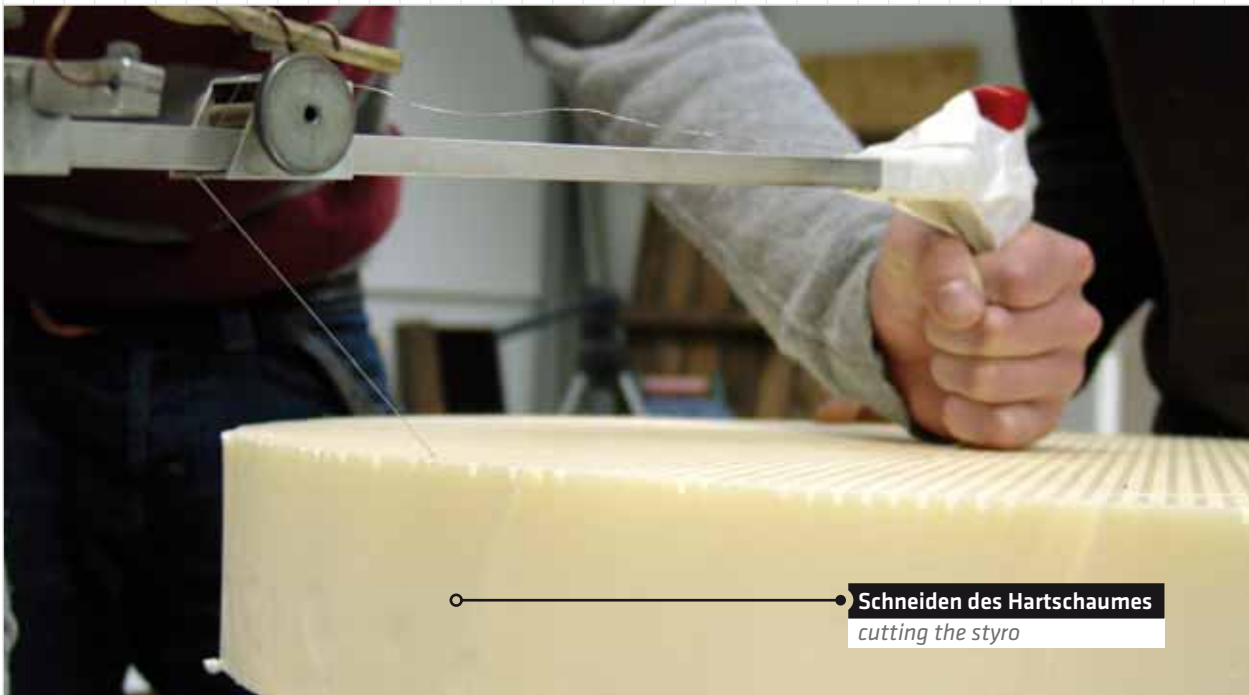
Umsetzung und Technik



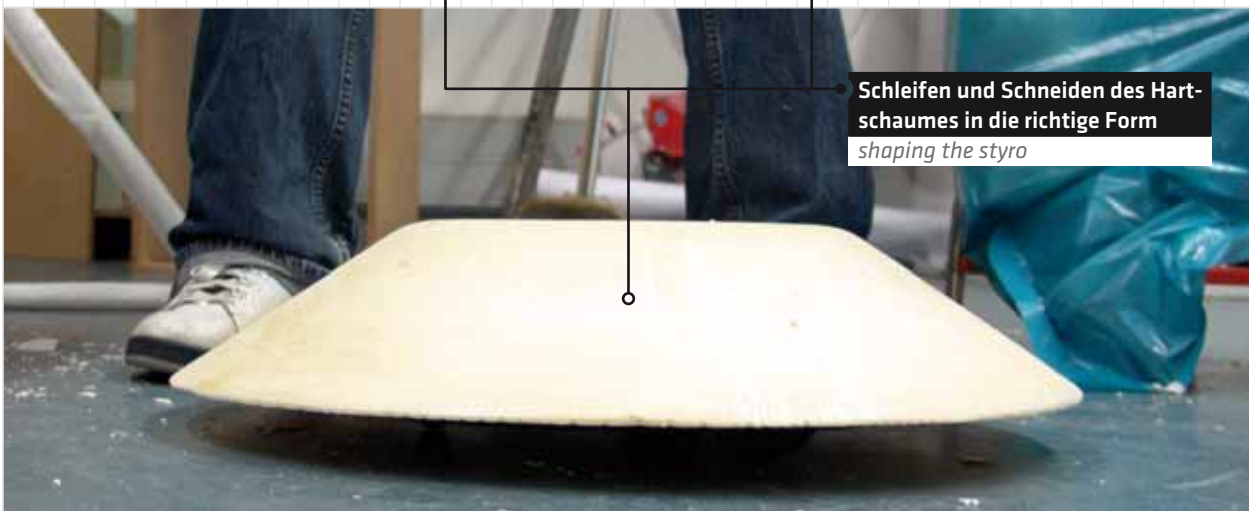
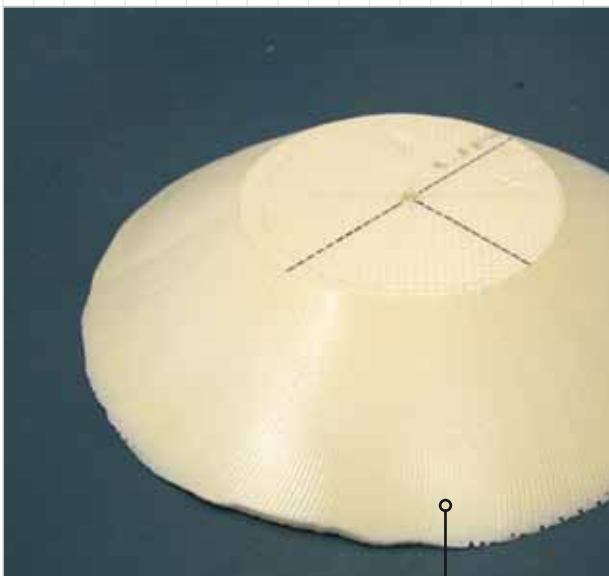
Umsetzung und Technik



Umsetzung und Technik

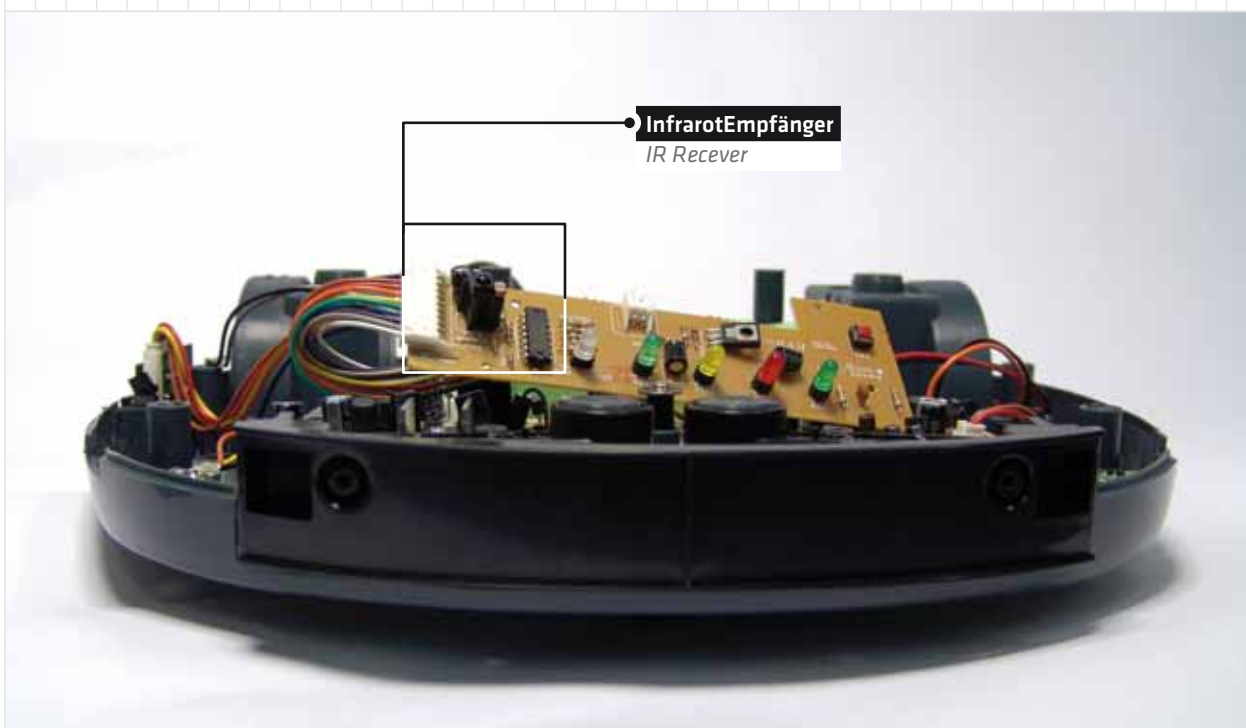
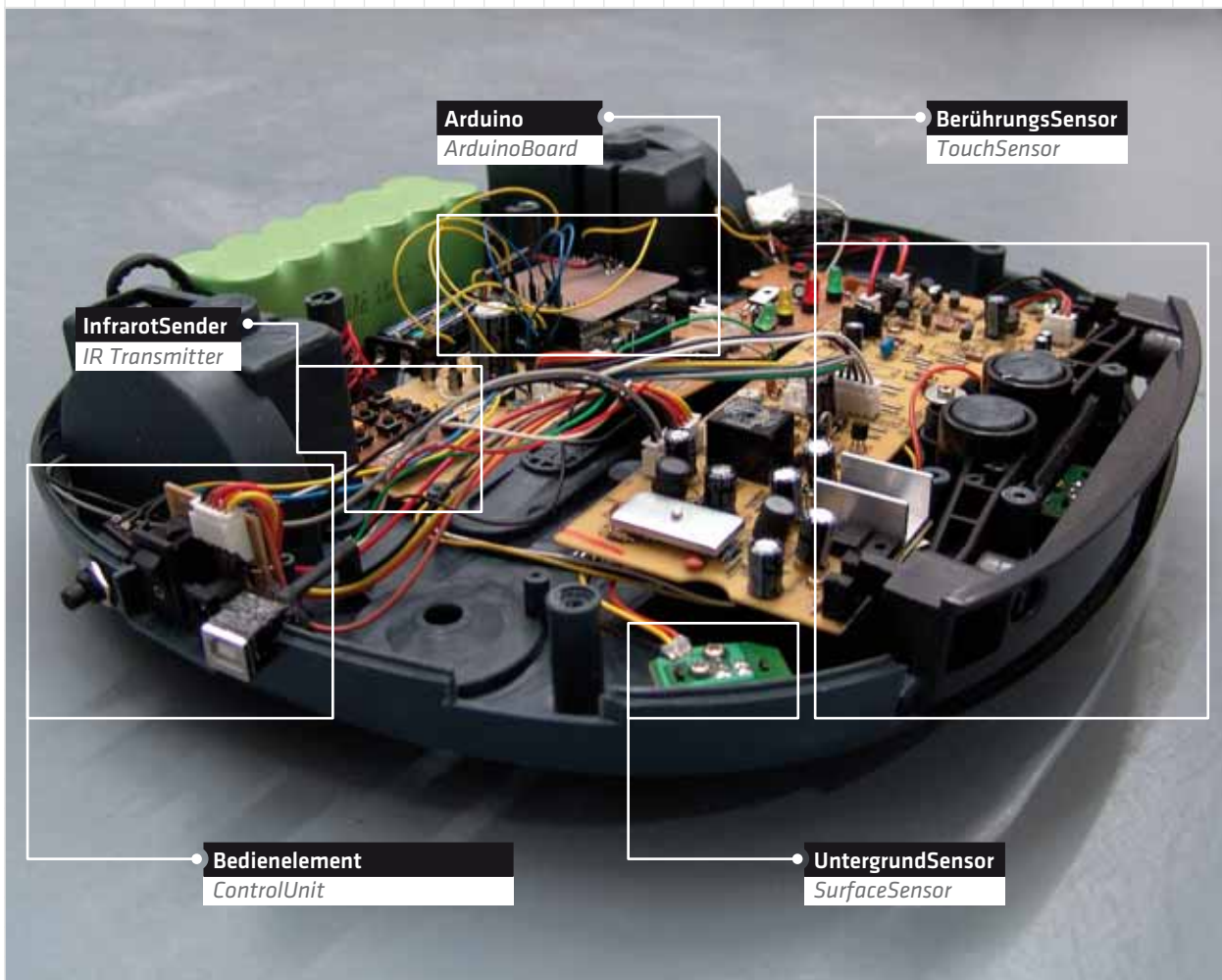


Schneiden des Hartschaumes
cutting the styro

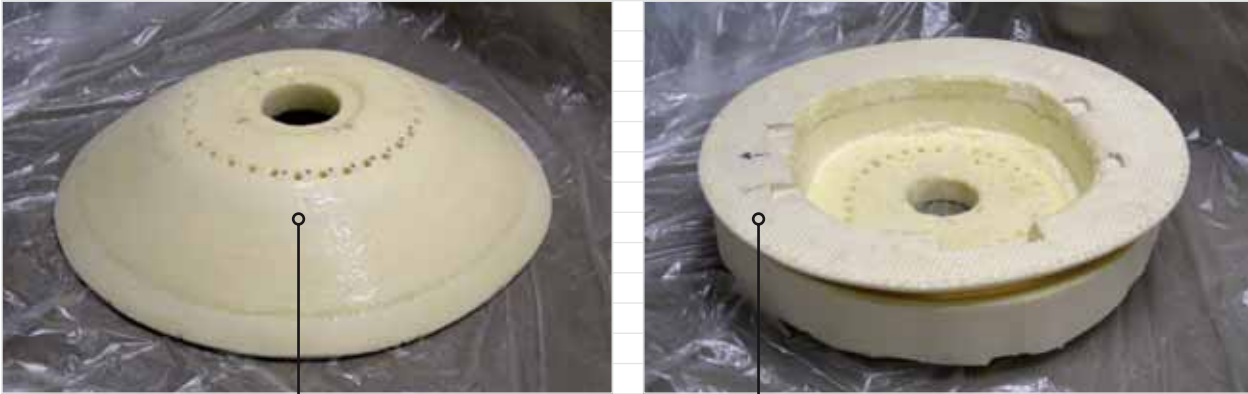


Schleifen und Schneiden des Hartschaumes in die richtige Form
shaping the styro

Umsetzung und Technik



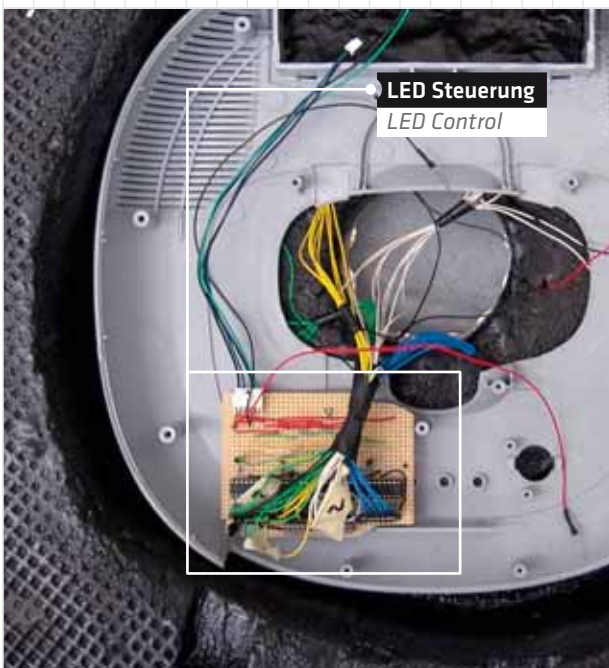
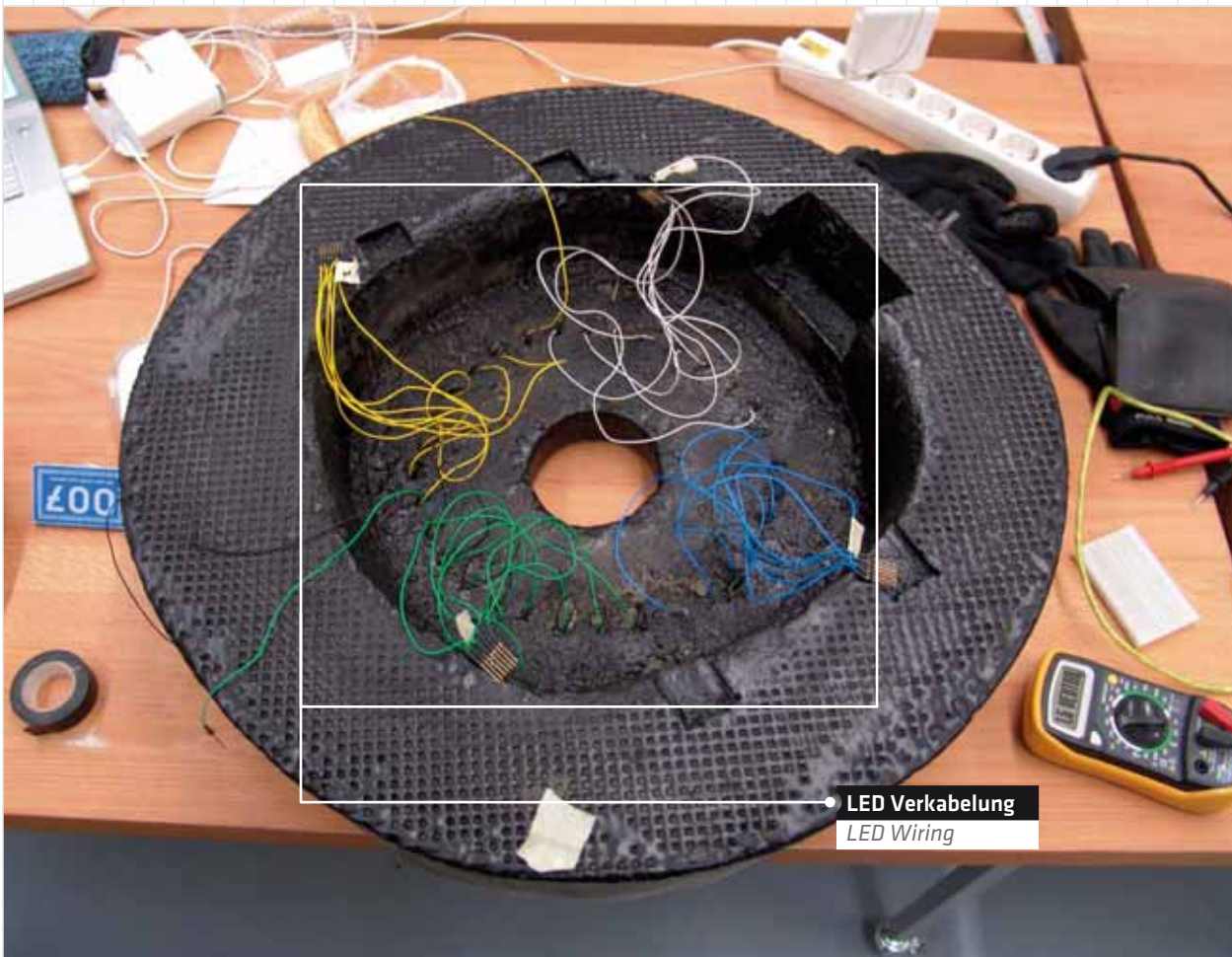
• Umsetzung und Technik



• EpoxydHarz gehärtete Form
Epoxy finished Shape



Umsetzung und Technik



Arduino Programmierung

Arduino Programmierung

```

//////////////////// LIGHT CONTROL
////////////////////
////////////////////

//Pin connected to ST_CP of 74HC595
int latchPin = 8;
//Pin connected to SH_CP of 74HC595
int clockPin = 12;
//Pin connected to DS of 74HC595
int dataPin = 11;

byte data = 0;

int lightmode;

long prevMillisLight = 0;
int interval;

//////////////////// PUT CONTROL
////////////////////
////////////////////
int inPin = 1; // choose the input pin
int putVal = 0; // variable for reading the putpin status
int threshold = 3;
boolean locked=false;

//////////////////// ROBO CONTROL
////////////////////
////////////////////

int inPoti = 5; // Potentiometer for modus selection
int potiVal;
int spiralPin = 2;
int autoPin = 3;
int stopPin = 5;
int roboMode = 0;
long prevMillisRobo;
int intervalRobo;
boolean autoFirstTime = true;
boolean autoRun = false;

boolean spiralFirstTime = true;
boolean spiralStop = false;

int LEDpin = 13;

////////////////////:-----SETUP

void setup() {
  lightmode = random(4);
  interval = random(5000,20000);
  intervalRobo= random(30000,50000);
  pinMode(LEDpin,OUTPUT);
  pinMode(latchPin, OUTPUT); ///// LIGHT

  pinMode(spiralPin,OUTPUT);
  pinMode(autoPin,OUTPUT);
  pinMode(stopPin,OUTPUT);

  Serial.begin(9600);
}

void switcher(int THE_PIN){
  digitalWrite(THE_PIN,HIGH);
  digitalWrite(LEDpin,HIGH);
  delay(200);
  digitalWrite(THE_PIN,LOW);
  digitalWrite(LEDpin,LOW);
}

void roboTimeUpdate(int THE_MIN, int THE_MAX){
  printString(„millisUpdate“);
  prevMillisRobo = millis();

  intervalRobo = random(THE_MIN,THE_MAX); // Stoptime
  printInteger(intervalRobo);
}

//////////////////// LOOP

void loop() {
  //.....putControl
  putVal = analogRead(inPin); // read input value
  potiVal = analogRead(inPoti);
  printInteger(putVal);
  printByte(10);
  printByte(13);
  if(potiVal <= 341){ ///// OFF Mode
    autoFirstTime = true;
    autoRun = false;
    spiralFirstTime = true;

    spiralStop = false;
    switcher(stopPin);
  }

  if(potiVal > 341 && potiVal <= 682){ ///// AutoMode
    spiralFirstTime = true;
    spiralStop = false;

    if(autoFirstTime == true){
      switcher(stopPin);
      printString(„autoSubroutineFirstTime“);
      roboTimeUpdate(5000,10000); ///// AutoInterval Wait First Time (30000,50000)
      autoFirstTime = false;
    }

    if(millis() - prevMillisRobo > intervalRobo){

      if(autoRun == false){
        switcher(autoPin);
        autoRun = true;
        roboTimeUpdate(5000,13000); ///// AutoInterval RunningTime

        printString(„AUTO“);
        printByte(10);
        printByte(13);
      }
      else{
        switcher(stopPin);
        autoRun = false;
        roboTimeUpdate(5000,10000); ///// AutoInterval Waiting Time (30000,50000)

        printString(„STOP“);
        printByte(10);
        printByte(13);
      }
    }

    if(potiVal > 682 && potiVal < 1024){ ///// Spiral Mode
      autoFirstTime = true;
      autoRun = false;

      if(spiralFirstTime == true){
        switcher(stopPin);
        printString(„spiralSubroutineFirstTime“);
        roboTimeUpdate(5000,5000); ///// SpiralMode First Waiting Time
        spiralFirstTime = false;
        spiralStop = false;
      }

      if(millis() - prevMillisRobo > intervalRobo){

        if(spiralStop == false){
          switcher(stopPin);
          spiralStop = true;
          roboTimeUpdate(3000,3000); ///// SpiralMode StopTime (at least 1000 millis)

          printString(„STOP“);
          printByte(10);
          printByte(13);
        }
        else{
          switcher(spiralPin);
          spiralStop = false;
          roboTimeUpdate(20000,20000); ///// SpiralMode Running Time

          printString(„SPIRAL“);
          printByte(10);
          printByte(13);
        }
      }

      //..... putCheck
      if (putVal > threshold && locked == false) { // check if the input is HIGH (button released)
        locked = true;
        lightmode = 6;
        prevMillisLight=millis();
        interval = 5000;
        printString(„put du hippie“);
        printByte(10);
        printByte(13);
      }

      if (putVal < threshold){

```

Arduino Programmierung

Arduino Programmierung

```

locked = false;
}

//.....lightControl

if(millis() - prevMillisLight > interval){
  prevMillisLight=millis();
  lightmode = random(5);
  interval = random(5000,20000);
}

if (lightmode == 6){
  blinkAll_2Bytes(1,50);
}

if (lightmode == 0){
  /// gegenläufig alle leds aussen, innen, aussen
  for (int j = 0; j <= 8; j++) {
    digitalWrite(latchPin, 0);
    lightShiftPinA(7-j);
    lightShiftPinA(j);
    digitalWrite(latchPin, 1);
    delay(50);
  }

  for (int j = 8; j >= 0; j--) {
    digitalWrite(latchPin, 0);
    lightShiftPinB(7-j);
    lightShiftPinB(j);
    digitalWrite(latchPin, 1);
    delay(50);
  }
}

if (lightmode == 1){
  // register 1 addieren dann register 2 subtrahieren, addieren
  dann register 1 subtrahieren
  for (int j = 0; j <= 16; j++) {
    digitalWrite(latchPin, 0);
    lightShiftPinA(15-j);
    lightShiftPinA(j);
    digitalWrite(latchPin, 1);
    delay(50);
  }

  for (int j = 16; j >= 0; j--) {
    digitalWrite(latchPin, 0);
    lightShiftPinB(15-j);
    lightShiftPinB(j);
    digitalWrite(latchPin, 1);
    delay(50);
  }
}

if (lightmode == 2){
  // innen nach aussen
  for (int j = 8; j >= 0; j--) {
    digitalWrite(latchPin, 0);
    lightShiftPinB(7-j);
    lightShiftPinB(j);
    digitalWrite(latchPin, 1);
    delay(50);
  }
}

if (lightmode == 3){
  // aussen nach innen
  for (int j = 0; j <= 8; j++) {
    digitalWrite(latchPin, 0);
    lightShiftPinB(7-j);
    lightShiftPinB(j);
    digitalWrite(latchPin, 1);
    delay(50);
  }
}

if (lightmode == 4){
  for (int j = 0; j <= 16; j++) {
    digitalWrite(latchPin, 0);
    lightShiftPinB(15-j);
    lightShiftPinB(j);
    digitalWrite(latchPin, 1);
    delay(50);
  }
}

if (lightmode == 5){
  for (int j = 16; j >= 0; j--) {
    digitalWrite(latchPin, 0);
    lightShiftPinB(15-j);
    lightShiftPinB(j);
    digitalWrite(latchPin, 1);
    delay(50);
  }
}

  delay(50);
}
}

int pin;

pin = 1<< p;

shiftOut(dataPin, clockPin, pin);
}

void lightShiftPinB(int p) {
  //defines a local variable
  int pin;

  pin = 1;

  for (int x = 0; x < p; x++) {
    pin = pin * 2;
  }
  shiftOut(dataPin, clockPin, pin);
}

// the program
void shiftOut(int myDataPin, int myClockPin, byte myDataOut) {

  //internal function setup
  int i=0;
  int pinState;
  pinMode(myClockPin, OUTPUT);
  pinMode(myDataPin, OUTPUT);

  digitalWrite(myDataPin, 0);
  digitalWrite(myClockPin, 0);
  for (i=7; i>=0; i--) {
    digitalWrite(myClockPin, 0);

    if ( myDataOut & (1<<i) ) {
      pinState= 1;
    }
    else {
      pinState= 0;
    }

    digitalWrite(myDataPin, pinState);
    digitalWrite(myClockPin, 1);
    digitalWrite(myDataPin, 0);
  }

  //stop shifting
  digitalWrite(myClockPin, 0);
}

//blinks both registers
void blinkAll_2Bytes(int n, int d) {
  digitalWrite(latchPin, 0);
  shiftOut(dataPin, clockPin, 0);
  shiftOut(dataPin, clockPin, 0);
  digitalWrite(latchPin, 1);
  delay(50);
  for (int x = 0; x < n; x++) {
    digitalWrite(latchPin, 0);
    shiftOut(dataPin, clockPin, 255);
    shiftOut(dataPin, clockPin, 255);
    digitalWrite(latchPin, 1);
    delay(d);
    digitalWrite(latchPin, 0);
    shiftOut(dataPin, clockPin, 0);
    shiftOut(dataPin, clockPin, 0);
    digitalWrite(latchPin, 1);
    delay(d);
  }
}

//////////Danke Frederic Gmeiner fürs Vorcoden //////////

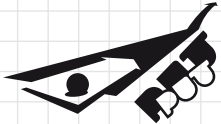
```

PUTPUT

Reclaim your urban environment playfully.

PUTPUT

Reclaim your urban environment
playfully.



Torsten Posselt

www.golb.vektorjunkies.net

Danke:

Frederic Gmeiner

Tim Horntrich

Benjamin Frenzel

Julius v. Bismark

Antje Hartenstein

Frau Bellmann



Universität der Künste Berlin

Enstanden im Wintersemester 06/07

Gestalten mit Digitalen Medien

Bereich Visuelle Kommunikation

Prof. Kora Kimpel

Dipl. Designer Dennis Paul