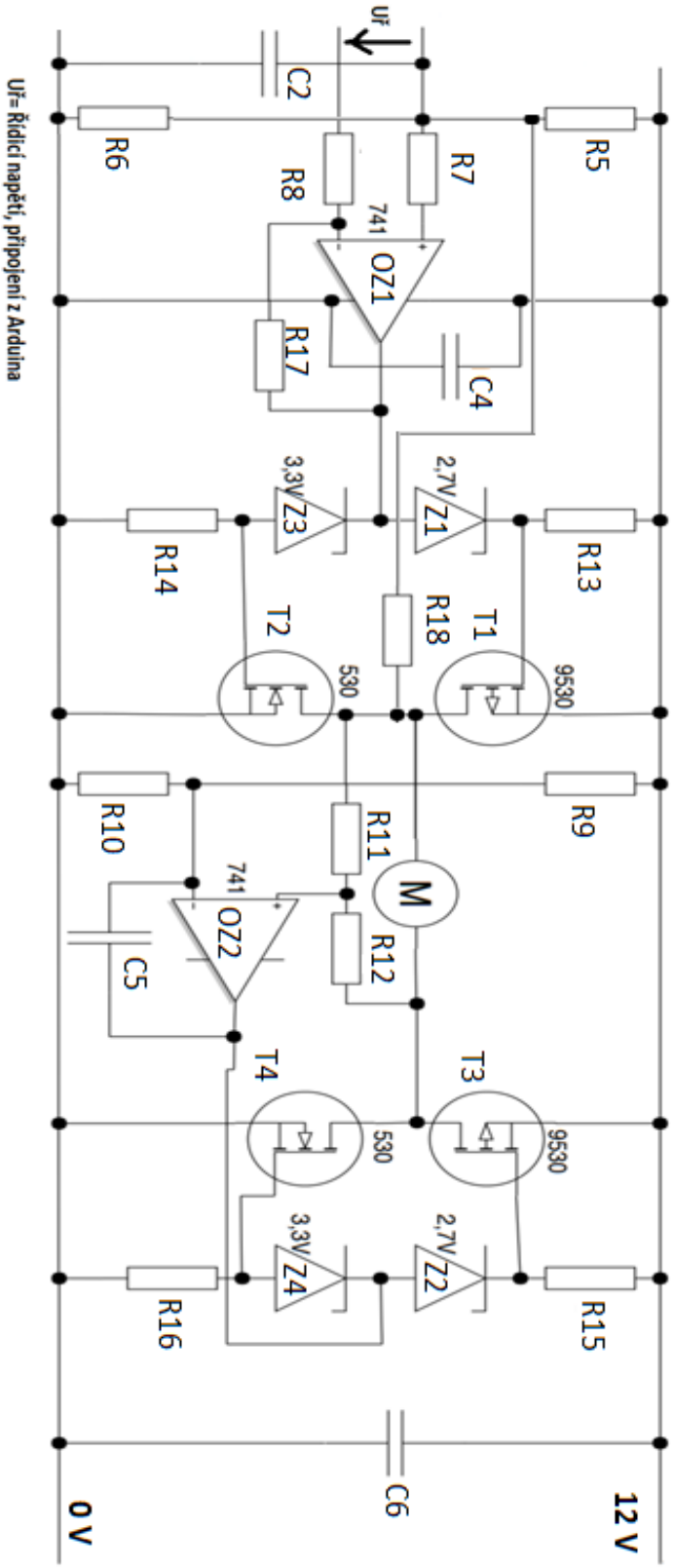
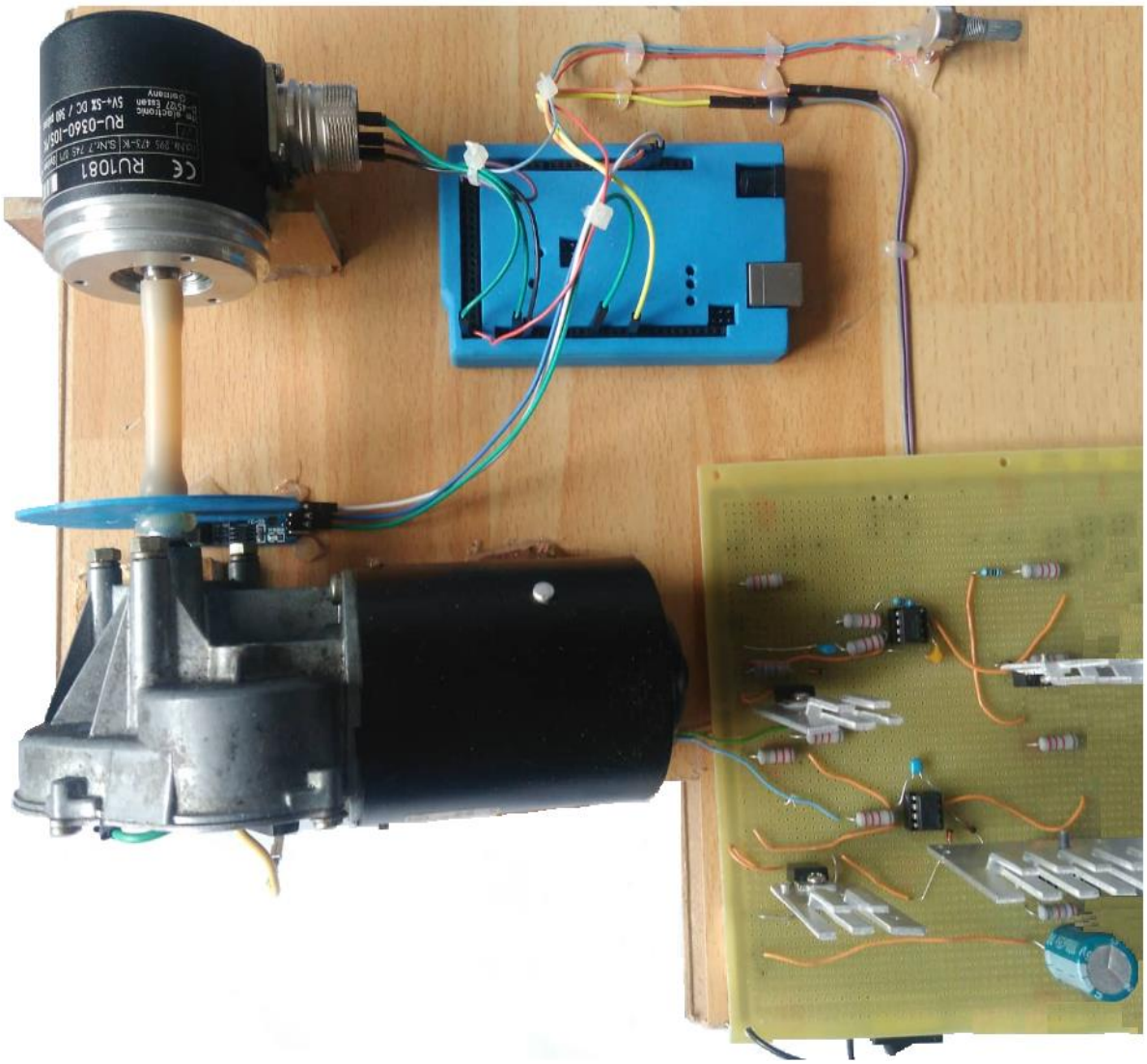
Přílohy

Příloha 1: Schéma zapojení výkonového obvodu



Příloha 2: Reálné zapojení všech součástí natáčení úhlu hřídele motoru



Příloha 3: Kód pro řízení polohy hřídele motoru na základě potenciometru

```
int motor = 7;
int potPin = A1;
int potentiometerValue = 0;
int calculatedIndex = 0;
int lastPotentiometer = 0;
int encoderPin1 = 18;
int encoderPin2 = 19;
int requiredPosition = 0;
volatile int lastEncoded = 0;
volatile long encoderValue = 0;
bool positionOK = false;

bool engineRun = true;

int positionArray
[]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22
,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,4
3,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84
,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,
104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,11
9,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,
135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,15
0,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,
166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,18
1,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,
197,198,199,191.....1192,1193,1194,1195,1196,1197,1198,1
199,1200,1201,1202,1203,1204,1205,1206,1207,1208,1209,1210,121
1,1212,1213,1214,1215,1216,1217,1218,1219,1220,1221,1222,1223,
1224,1225,1226,1227,1228,1229,1230,1231,1232,1233,1234,1235,12
36,1237,1238,1239,1240,1241,1242,1243,1244,1245,1246,1247,1248
,1249,1250,1251,1252,1253,1254,1255,1256,1257,1258,1259,1260,1
261,1262,1263,1264,1265,1266,1267,1268,1269,1270,1271,1272,127
3,1274,1275,1276,1277,1278,1279,1280,1281,1282,1283,1284,1285,
1286,1287,1288,1289,1290,1291,1292,1293,1294,1295,1296,1297,12
98,1299,1300,1301,1302,1303,1304,1305,1306,1307,1308,1309,1310
,1311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321,1322,1
323,1324,1325,1326,1327,1328,1329,1330,1331,1332,1333,1334,133
5,1336,1337,1338,1339,1340,1341,1342,1343,1344,1345,1346,1347,
1348,1349,1350,1351,1352,1353,1354,1355,1356,1357,1358,1359,13
60,1361,1362,1363,1364,1365,1366,1367,1368,1369,1370,1371,1372
,1373,1374,1375,1376,1377,1378,1379,1380,1381,1382,1383,1384,1
385,1386,1387,1388,1389,1390,1391,1392,1393,1394,1395,1396,139
7,1398,1399,1400,1401,1402,1403,1404,1405,1406,1407,1408,1409,
1410,1411,1412,1413,1414,1415,1416,1417,1418,1419,1420,1421,14
22,1423,1424,1425,1426,1427,1428,1429,1430,1431,1432,1433,1434
,1435,1436,1437,1438,1439,1440,1441,1442,1443,1444,1445,1446,1
447,1448,1449,1450};
```

```
void setup()
{
     pinMode(encoderPin1, INPUT_PULLUP);
     pinMode(encoderPin2, INPUT_PULLUP);
     pinMode(motor, OUTPUT);
     digitalWrite(encoderPin1, HIGH);
     digitalWrite(encoderPin2, HIGH);
     attachInterrupt(0, referencePosition, RISING);
     while(positionOK == false
          {
          analogWrite(motor,190);
     }
     delay(1000);
     attachInterrupt(4, updateEncoder, CHANGE);
     attachInterrupt(5, updateEncoder, CHANGE);
     encoderValue = 0;
}

void loop()
{
     potentiometerValue = analogRead(potPin);
     if (lastPotentiometer != potentiometerValue)
     {
          engineRun = true;
          lastPotentiometer = potentiometerValue;
          calculatedIndex = map(lastPotentiometer, 0, 1023, 0,
1450);
          requiredPosition = positionArray[calculatedIndex];
          if( requiredPosition < encoderValue)
```

```
            analogWrite(motor,50);
            if( requiredPosition > encoderValue)
            analogWrite(motor,200);
            while(engineRun)
            {
                  if(encoderValue == requiredPosition)
                  {
                        analogWrite(motor,107);
                        engineRun = false;
                        delay(1000);
                  }
            }
      }
}
void referencePosition()
{
      analogWrite(motor,105);
      positionOK = true;
      detachInterrupt(0);
void updateEncoder()
{
      int MSB = digitalRead(encoderPin1);
      int LSB = digitalRead(encoderPin2);
      int encoded = (MSB << 1) |LSB;
      int sum = (lastEncoded << 2) | encoded;
      if(sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum
      == 0b1011) encoderValue ++;
      if(sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum
      == 0b1000) encoderValue --;
      lastEncoded = encoded;
```