**HW2 Optical Tachometer**

Johns Hopkins University

Real Time Software for Embedded Systems

Fall 2014

Tony Florida

2014-09-24

**Requirements**

Hardware

* There shall be an IR led emitter and detector circuit
* There shall be a motor that is spun via electronic speed control (ESC)
* An Arduino board shall control the circuits
* External battery supply shall be used for the motor
* Blades shall be attached to the motor and pass through the emitter/detector pair

Software

* The software running on the Arduino shall use function queue scheduling design
* The software shall capture the rotations per minute (RPM) every second
* The software shall accept serial commands to control the ESC

Test

* The test shall issue various serial commands to the ESC to vary the speed of the blades
* The blades shall be attached to the motor and pass through the emitter/detector pair

Parts List

* (1) Arduino Uno
* (1) 10k resistor
* (1) 220 ohm resistor
* (1) spool of hobby wire
* (1) USB 2.0 A/B cable
* (1) breadboard
* (1) 12v battery
* (1) LED of any color
* (1) Infrared Emitter and Detector (Radio shack 276-142)
* (1) Turnigy Multistar 30 Amp Multi-rotor Brushless ESC 2-4S
* (1) Gemfan 9x4.7 Nylon Prop Set (1x CW & 1x CCW)
* (1) Turnigy Multistar ESC Programming Card
* Wood to hold motor and emitter/detector
* Nails
* Twist ties
* Black electrical tape

Required Software

* Arduino Sketch v1.0
* Microsoft Excel 2010

**Architecture**

Hardware

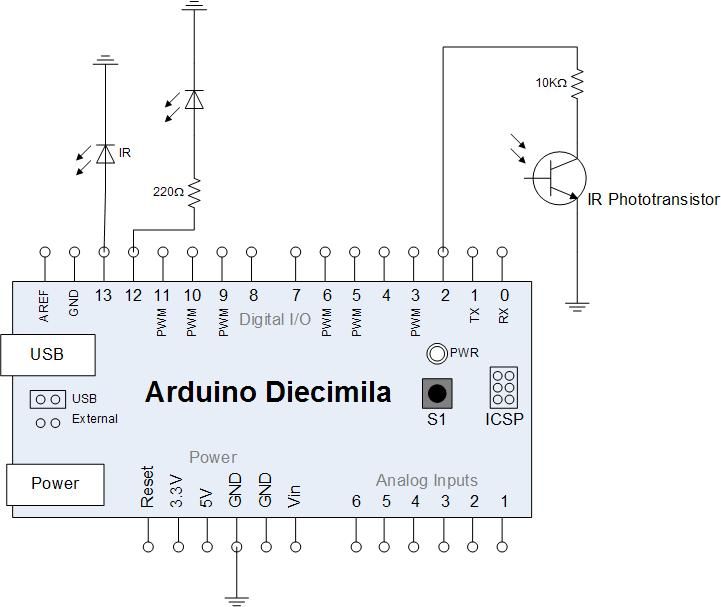


Figure - Circuit Schematic [1]

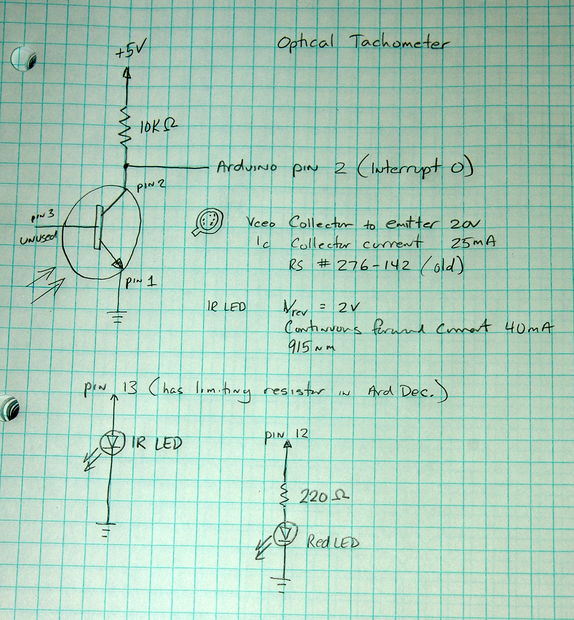


Figure – Schematic Notes[1]

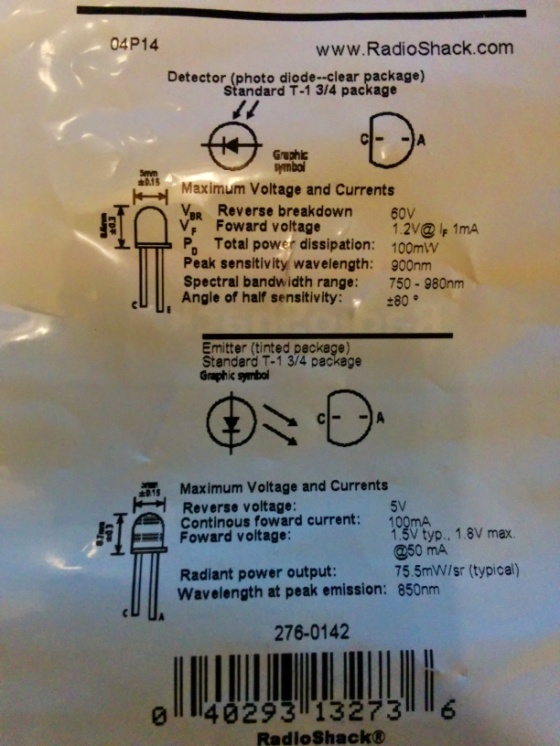


Figure – Emitter Detector Sensors

Software



Figure - Software Architecture Diagram

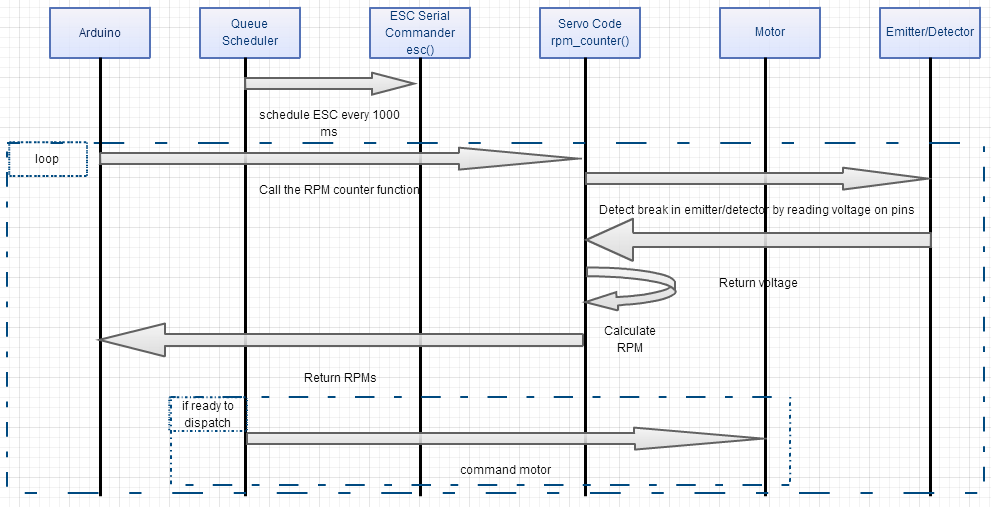


Figure - Hardware/Software Sequence Diagram

**Design**

Software



Photos of the Hardware

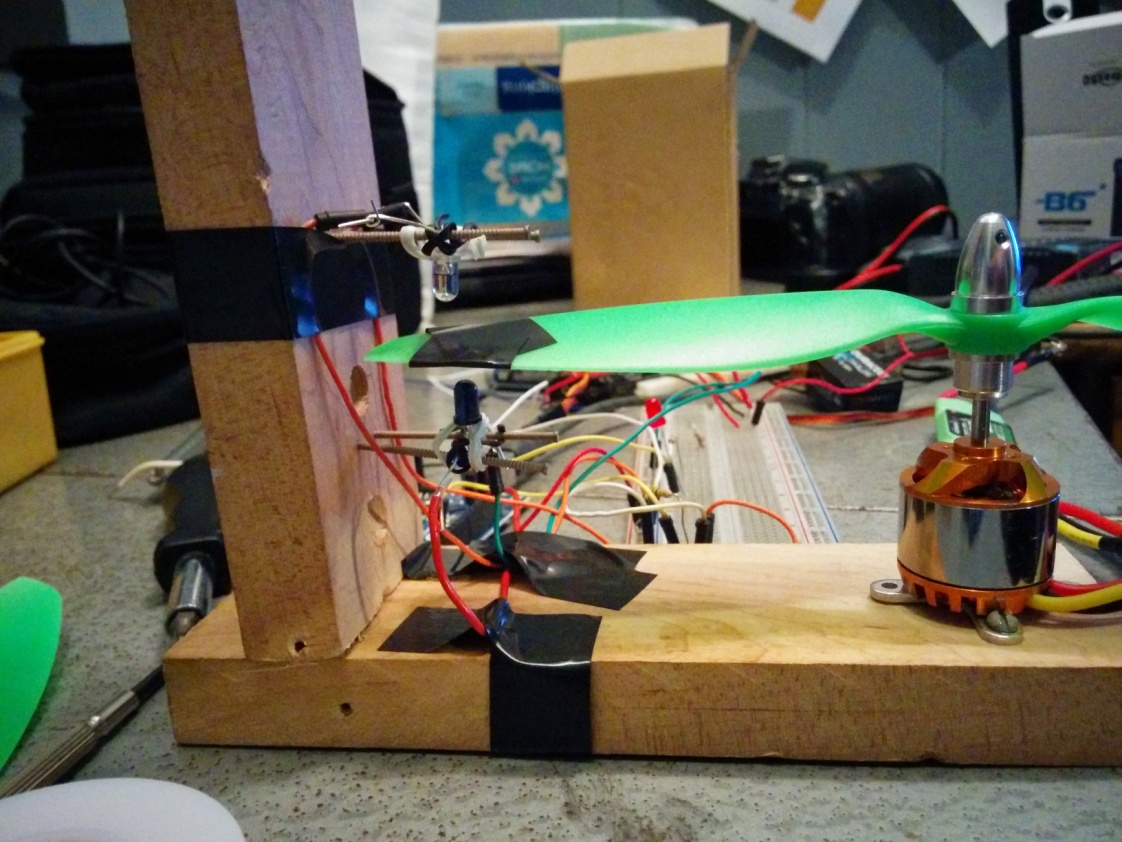


Figure – Side View of the Test Setup

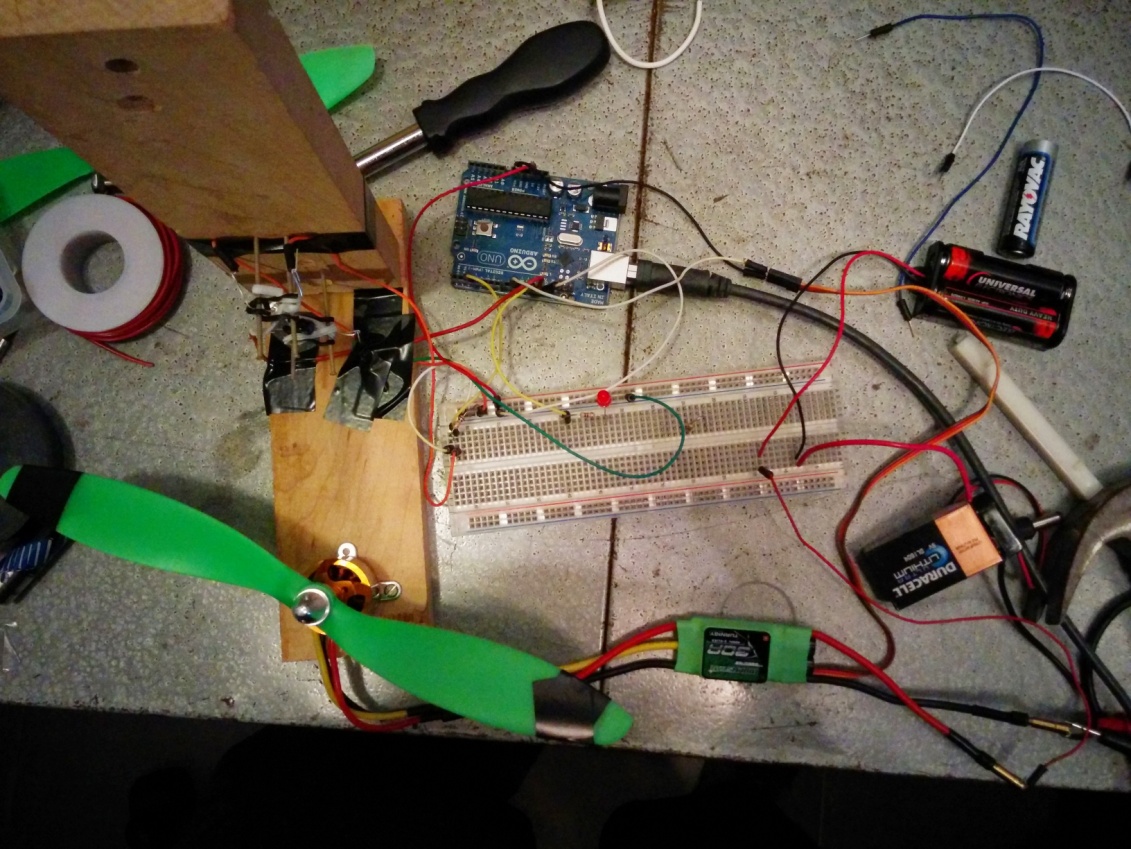


Figure - Top View of the Test Setu

**Implementation**

**#include <Queue.h>**

**#include <Servo.h>**

**//JHU RTSW HW 2 - Optical Tachometer**

**//Tony Florida**

**//2014-09-24**

**//References:**

**// http://www.instructables.com/id/Arduino-Based-Optical-Tachometer/**

**// http://techvalleyprojects.blogspot.com/2012/06/arduino-control-escmotor-tutorial.html**

**// https://github.com/Zuph/AVRQueue**

**//ESC variables**

**// This is our motor.**

**Servo myMotor;**

**// This is the final output**

**// written to the motor.**

**String incomingString;**

**//IR Emitter Detector variables**

**int ledPin = 13; // IR LED connected to digital pin 13**

**int statusPin = 12; // LED connected to digital pin 12**

**volatile byte rpmcount;**

**volatile int status;**

**unsigned int rpm;**

**unsigned long timeold;**

**//IR Emitter Detector function**

**void rpm\_fun()**

**{**

**//Each rotation, this interrupt function is run twice, so take that into consideration for**

**//calculating RPM**

**//Update count**

**rpmcount++;**

**//Toggle status LED**

**if (status == LOW) {**

**status = HIGH;**

**} else {**

**status = LOW;**

**}**

**digitalWrite(statusPin, status);**

**}**

**void setup() {**

**// Required for I/O from Serial monitor**

**Serial.begin(9600);**

**//ESC setup**

**Serial.println("Initializing ESC");**

**// Put the motor to Arduino pin #9**

**myMotor.attach(9);**

**//IR Emitter Detector setup**

**//Interrupt 0 is digital pin 2, so that is where the IR detector is connected**

**//Triggers on FALLING (change from HIGH to LOW)**

**attachInterrupt(0, rpm\_fun, FALLING);**

**//Turn on IR LED**

**pinMode(ledPin, OUTPUT);**

**digitalWrite(ledPin, HIGH);**

**//Use statusPin to flash along with interrupts**

**pinMode(statusPin, OUTPUT);**

**rpmcount = 0;**

**rpm = 0;**

**timeold = 0;**

**status = LOW;**

**//Function queue scheduling setup**

**Serial.println("Initializing function queue scheduling");**

**Queue myQueue;**

**myQueue.scheduleFunction(esc, "ESC", 5000, 1000);**

**while(1) {**

**myQueue.Run(millis());**

**rpm\_counter();**

**}**

**}**

**//Receive ESC commands via serial**

**int esc(unsigned long now)**

**{**

**// If there is incoming value**

**if(Serial.available() > 0)**

**{**

**// read the value**

**char ch = Serial.read();**

**/\***

**\* If ch isn't a newline**

**\* (linefeed) character,**

**\* we will add the character**

**\* to the incomingString**

**\*/**

**if (ch != 10){**

**// Print out the value received**

**// so that we can see what is**

**// happening**

**//Serial.print("I have received: ");**

**//Serial.print(ch, DEC);**

**//Serial.print('\n');**

**// Add the character to**

**// the incomingString**

**incomingString += ch;**

**}**

**// received a newline (linefeed) character**

**// this means we are done making a string**

**else**

**{**

**// print the incoming string**

**//Serial.println("I am printing the entire string");**

**//Serial.println(incomingString);**

**// Convert the string to an integer**

**int val = incomingString.toInt();**

**// print the integer**

**//Serial.println("Printing the value: ");**

**//Serial.println(val);**

**/\***

**\* We only want to write an integer between**

**\* 0 and 180 to the motor.**

**\*/**

**if (val > -1 && val < 181)**

**{**

**// Print confirmation that the**

**// value is between 0 and 180**

**//Serial.println("Value is between 0 and 180");**

**// Write to Servo**

**myMotor.write(val);**

**}**

**// The value is not between 0 and 180.**

**// We do not want write this value to**

**// the motor.**

**else**

**{**

**//Serial.println("Value is NOT between 0 and 180");**

**// IT'S a TRAP!**

**//Serial.println("Error with the input");**

**} //0 to 180**

**// Reset the value of the incomingString**

**incomingString = "";**

**} //ch not 10**

**}**

**}**

**//Count RPMs**

**void rpm\_counter()**

**{**

**//Update RPM every second**

**delay(1000);**

**//Don't process interrupts during calculations**

**detachInterrupt(0);**

**//Note that this would be 60\*1000/(millis() - timeold)\*rpmcount if the interrupt**

**//happened once per revolution instead of twice. Other multiples could be used**

**//for multi-bladed propellers or fans**

**rpm = 30\*1000/(millis() - timeold)\*rpmcount;**

**timeold = millis();**

**rpmcount = 0;**

**//Write it out to serial port**

**Serial.print(millis());**

**Serial.print(" ");**

**Serial.println(rpm,DEC);**

**//Restart the interrupt processing**

**attachInterrupt(0, rpm\_fun, FALLING);**

**}**

**//not using the loop in this program**

**void loop() {**

**}**

**Results**

Log

1000 0

2000 0

3001 0

4001 0

5002 0

6002 0

7003 0

8003 0

9004 0

10004 0

11004 0

12006 0

13006 0

14007 0

15007 0

16008 0

17008 0

18009 0

19009 0

20009 0

21010 0

22011 0

23012 0

24012 0

25013 0

26013 0

27014 0

28014 0

29015 0

30015 0

31016 0

32017 0

33017 0

34018 0

35018 0

36019 0

37019 0

38020 0

39020 0

40020 0

41022 0

42022 0

43023 0

44023 0

45024 0

46024 0

47025 0

48025 0

49026 0

50026 0

51026 0

52028 0

53028 0

54029 0

55029 0

56030 0

57030 0

58031 0

59031 0

60032 0

61032 0

62032 0

63033 0

64034 0

65035 0

66035 0

67036 0

68036 0

69037 0

70037 0

71037 0

72038 0

73038 0

74039 0

75040 0

76041 0

77041 0

78042 0

79042 0

80043 0

81043 0

82043 0

83044 0

84044 0

85045 0

86046 0

87047 0

88047 0

89048 0

90048 0

91048 0

92049 0

93049 0

94050 0

95050 0

96052 0

97052 0

98053 0

99053 0

100054 0

101054 0

102054 0

103055 0

104055 0

105057 0

106057 0

107058 0

108058 0

109059 0

110059 0

111059 0

112061 0

113061 0

114062 0

115062 0

116063 0

117063 0

118064 0

119065 0

120066 0

121066 0

122066 0

123067 0

124067 0

125068 0

126069 0

127070 0

128070 0

129071 0

130071 0

131072 0

132072 0

133073 0

134074 0

135074 0

136075 0

137075 0

138076 0

139076 0

140078 0

141078 0

142078 0

143079 0

144079 0

145080 0

146080 0

147082 0

148082 0

149083 0

150083 0

151084 0

152084 0

153084 0

154086 0

155086 0

156087 0

157087 0

158088 0

159088 0

160089 0

161089 0

162091 0

163091 0

164091 0

165092 0

166092 0

167093 0

168093 0

169095 0

170095 0

171096 0

172096 0

173096 0

174097 0

175097 0

176099 0

177099 0

178100 0

179100 0

180101 0

181101 0

182102 0

183103 0

184103 0

185104 0

186104 0

187105 0

188106 0

189107 0

190107 0

191108 0

192108 0

193108 0

194110 0

195110 0

196111 0

197111 870

198112 667

199113 540

200114 522

201114 510

202115 522

203115 510

204116 522

205117 493

206117 510

207118 493

208119 493

209120 493

210120 510

211121 493

212122 493

213123 493

214123 510

215123 540

216125 493

217125 510

218126 493

219126 510

220128 464

221128 2220

222129 2523

223130 2580

224131 2436

225131 2520

226131 2490

227133 2378

228133 2460

229134 2349

230135 2349

231136 2349

232136 2400

233137 2320

234138 2291

235139 2291

236139 2340

237140 2291

238141 2262

239141 2340

240143 2233

241143 2310

242144 2204

243145 2310

244146 2204

245146 2280

246147 2175

247148 2204

248148 2250

249149 2175

250150 2146

251151 2146

252151 2190

253152 2146

254153 2117

255154 2146

256154 2190

257156 2117

258156 2160

259156 2160

260158 2088

261158 2160

262159 2088

263160 2160

264161 2088

265161 2130

266162 2088

267163 2059

268164 2088

269164 2130

270166 2088

271166 2130

272166 2160

273167 2059

274168 2059

275169 2088

276169 2130

277171 2088

278171 2130

279172 2088

280172 2130

281174 2059

282174 2160

283174 2130

284176 2059

285176 2130

286177 2059

287178 2130

288179 2059

289179 2130

290180 2059

291181 2030

292182 2059

293182 2130

294183 2030

295184 2059

296184 2100

297185 2059

298186 2030

299187 2059

300187 2100

301189 2030

302189 2130

303190 2059

304190 2100

305191 2059

306192 2030

307192 2100

308194 2030

309194 2100

310195 2030

311196 2100

312197 2030

313197 2070

314198 2001

315199 2059

316199 2070

317200 2030

318201 2030

319202 2001

320202 2070

321203 2030

322204 2001

323205 2001

324205 2100

325207 2001

326207 2070

327207 2100

328209 2001

329209 2100

330210 2001

331210 2070

332212 2030

333212 2070

334213 2001

335214 2001

336215 2001

337215 2100

338215 2070

339217 2001

340217 2070

341218 2030

342219 2001

343220 2001

344220 2070

345221 2030

346222 2001

347223 2001

348223 2070

349224 2030

350225 2001

351225 2070

352226 2030

353227 2001

354228 2001

355228 2100

356230 2030

357230 2070

358231 2001

359232 2001

360232 2100

361233 2001

362233 2070

363235 2001

364235 2070

365236 2001

366237 2001

367238 2030

368238 2070

369239 1972

370240 2001

371240 2070

372241 2001

373242 2001

374243 1421

375243 660

376245 464

377245 480

378246 435

379246 480

380248 464

381248 450

382248 480

383249 464

384250 480

385251 435

386251 480

387252 464

388252 480

389254 464

390254 480

391255 464

392255 480

393256 464

394257 464

395257 150

396259 0

397259 0

398260 0

399260 0

400261 0

401261 0

402262 0

403263 0

404263 0

405264 0

406264 0

407265 0

408265 0

Plot

Video Presentation

<https://www.youtube.com/watch?v=vAZy4O3XNxI>

**References**

[1] <http://www.instructables.com/id/Arduino-Based-Optical-Tachometer/>

[2] <http://techvalleyprojects.blogspot.com/2012/06/arduino-control-escmotor-tutorial.html>

[3] <https://github.com/Zuph/AVRQueue>