**Appendix 4: Assembly Program**

1 @DATA

2 offset DS 1

3 stackPointer DS 1

4 outputs DS 12

5 state DS 1

6

7 @CODE

8

9 TIMEMOTORDOWN **EQU** 150

10 BELT **EQU** 2000

11 BELTROUND **EQU** 2000

12 SORT **EQU** 200

13 COUNTDOWN **EQU** 30000

14 LENSLAMPPOSITION **EQU** 2

15 LENSLAMPSORTER **EQU** 6

16 HBRIDGE0 **EQU** 0

17 HBRIDGE1 **EQU** 1

18 CONVEYORBELT **EQU** 7

19 FEEDERENGINE **EQU** 3

20 DISPLAY **EQU** 8

21 LEDSTATEINDICATOR **EQU** 9

22 begin: BRA main

23

24

25 *;sleep*

26 \_timer: MULS R5 10

27 PUSH R4

28 LOAD R4 R5

29 LOAD R5 -16

30 LOAD R5 [R5+13]

31 SUB R5 R4

32 LOAD R4 -16

33 \_wait: CMP R5 [R4+13] *; Compare the timer to 0*

34 BMI \_wait

35 PULL R4

36 RTS

37

38 \_pressed: PUSH R4 *;make sure all vars are the same at the end*

39 PUSH R5

40 LOAD R4 R3

41 LOAD R5 2

42 BRS \_pow

43 LOAD R3 R5

44 LOAD R5 -16

45 LOAD R4 [R5+7]

46 DIV R4 R3

47 MOD R4 2

48

49 PUSH R4 *;the result*

50 ADD SP 1 *;decrease the SP so we get the correct pulls*

51

52 PULL R5

53 PULL R4

54

55 RTS

56

57 \_pow: CMP R4 0

58 BEQ \_pow1

59 CMP R4 1

60 BEQ \_powR

61 PUSH R3

62 PUSH R4

63 SUB R4 1

64 LOAD R3 R5

65 \_powLoop: MULS R5 R3

66 SUB R4 1

67 CMP R4 0

68 BEQ \_powReturn

69 BRA \_powLoop

70 \_powReturn: PULL R4

71 PULL R3

72 RTS

73 \_pow1: LOAD R5 1

74 RTS

75 \_powR: RTS

76

77 *;display*

78 \_Hex7Seg: BRS \_Hex7Seg\_bgn *; push address(tbl) onto stack and proceed at bgn*

79 \_Hex7Seg\_tbl: CONS %01111110 *; 7-segment pattern for '0'*

80 CONS %00110000 *; 7-segment pattern for '1'*

81 CONS %01101101 *; 7-segment pattern for '2'*

82 CONS %01111001 *; 7-segment pattern for '3'*

83 CONS %00110011 *; 7-segment pattern for '4'*

84 CONS %01011011 *; 7-segment pattern for '5'*

85 CONS %01011111 *; 7-segment pattern for '6'*

86 CONS %01110000 *; 7-segment pattern for '7'*

87 CONS %01111111 *; 7-segment pattern for '8'*

88 CONS %01111011 *; 7-segment pattern for '9'*

89 CONS %01110111 *; 7-segment pattern for 'A'*

90 CONS %00011111 *; 7-segment pattern for 'b'*

91 CONS %01001110 *; 7-segment pattern for 'C'*

92 CONS %00111101 *; 7-segment pattern for 'd'*

93 CONS %01001111 *; 7-segment pattern for 'E'*

94 CONS %01000111 *; 7-segment pattern for 'F'*

95 \_Hex7Seg\_bgn: AND R5 %01111 *; R0 = R0 MOD 16 , just to be safe...*

96 LOAD R4 [SP++] *; R4 = address(tbl) (retrieve from stack)*

97 LOAD R4 [R4+R5] *; R4 = tbl[R0]*

98 LOAD R5 -16

99 STOR R4 [R5+8] *; and place this in the Display Element*

100 RTS

101 main: STOR R5 [GB +offset + 0] *;storeData(R5, 'offset', 0)*

102 LOAD R0 timerInterrupt *;installCountdown('timerInterrupt')*

103 ADD R0 R5

104 LOAD R1 16

105 STOR R0 [R1]

106

107 LOAD R5 -16

108

109 *; Set the timer to 0*

110 LOAD R0 0

111 SUB R0 [R5+13]

112 STOR R0 [R5+13]

113 STOR SP [GB +stackPointer + 0] *;storeData(SP, 'stackPointer', 0)*

114 LOAD R0 0 *;$counter = 0*

115 LOAD R1 0 *;$location = 0*

116 LOAD R2 0 *;$sleep = 0*

117 LOAD R3 0 *;$temp = 0*

118 STOR R3 [GB +outputs + HBRIDGE1] *;storeData($temp, 'outputs', HBRIDGE1)*

119 STOR R3 [GB +outputs + LENSLAMPPOSITION] *;storeData($temp, 'outputs', LENSLAMPPOSITION)*

120 STOR R3 [GB +outputs + LENSLAMPSORTER] *;storeData($temp, 'outputs', LENSLAMPSORTER)*

121 STOR R3 [GB +outputs + LEDSTATEINDICATOR] *;storeData($temp, 'outputs', LEDSTATEINDICATOR)*

122 STOR R3 [GB +outputs + DISPLAY] *;storeData($temp, 'outputs', DISPLAY)*

123 STOR R3 [GB +outputs + CONVEYORBELT] *;storeData($temp, 'outputs', CONVEYORBELT)*

124 STOR R3 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

125 LOAD R4 0 *;$state = 0*

126 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

127 LOAD R3 10 *;$temp = 10*

128 STOR R3 [GB +outputs + HBRIDGE0] *;storeData($temp, 'outputs', HBRIDGE0)*

129 *;unset($temp, $state)*

130 BRA initial *;initial()*

131

132 timerInterrupt: BRS timerManage *;timerManage()*

133 LOAD R3 5 *;$temp = 5*

134 PUSH R5 *;display($temp, 'display')*

135 PUSH R4

136 LOAD R5 R3

137 BRS \_Hex7Seg

138 LOAD R4 %0000001

139 STOR R4 [R5+9]

140 PULL R4

141 PULL R5

142 LOAD R3 10 *;$temp = 10*

143 STOR R3 [GB +outputs + HBRIDGE0] *;storeData($temp, 'outputs', HBRIDGE0)*

144 LOAD R3 0 *;$temp = 0*

145 STOR R3 [GB +outputs + LENSLAMPPOSITION] *;storeData($temp, 'outputs', LENSLAMPPOSITION)*

146 STOR R3 [GB +outputs + LENSLAMPSORTER] *;storeData($temp, 'outputs', LENSLAMPSORTER)*

147 STOR R3 [GB +outputs + LEDSTATEINDICATOR] *;storeData($temp, 'outputs', LEDSTATEINDICATOR)*

148 STOR R3 [GB +outputs + DISPLAY] *;storeData($temp, 'outputs', DISPLAY)*

149 STOR R3 [GB +outputs + CONVEYORBELT] *;storeData($temp, 'outputs', CONVEYORBELT)*

150 STOR R3 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

151 PUSH R5 *;display($temp, 'display')*

152 PUSH R4

153 LOAD R5 R3

154 BRS \_Hex7Seg

155 LOAD R4 %0000001

156 STOR R4 [R5+9]

157 PULL R4

158 PULL R5

159 *;unset($temp)*

160 LOAD R3 [ GB + offset + 0 ] *;$temp = getData('offset', 0)*

161 LOAD R4 initial *;$temp2 = getFuncLocation('initial')*

162 ADD R3 R4 *;$temp += $temp2*

163 ADD SP 2 *;addStackPointer(2)*

164 PUSH R3 *;pushStack($temp)*

165 ADD SP -1 *;addStackPointer(-1)*

166 RTE

167

168 initial: LOAD R3 0 *;$temp = 0*

169 PUSH R5 *;display($temp, 'leds2')*

170 LOAD R5 -16

171 STOR R3 [R5+10]

172 PULL R5

173 LOAD R3 [ GB + stackPointer + 0 ] *;$temp = getData('stackPointer', 0)*

174 LOAD SP R3 *;setStackPointer($temp)*

175 BRS timerManage *;timerManage()*

176 PUSH R3 *;$push = getButtonPressed(5)*

177 LOAD R3 5

178 BRS \_pressed

179 PULL R3

180 SUB SP 5

181 PULL R4

182 ADD SP 4

183 CMP R4 1 *;if ($push == 1) {*

184 BEQ conditional0

185 return0: *;unset($push)*

186 BRA initial *;initial()*

187

188 *;if ($push == 1) {*

189 conditional0: LOAD R3 0 *;$temp = 0*

190 STOR R3 [GB +outputs + HBRIDGE0] *;storeData($temp, 'outputs', HBRIDGE0)*

191 LOAD R3 10 *;$temp = 10*

192 STOR R3 [GB +outputs + HBRIDGE1] *;storeData($temp, 'outputs', HBRIDGE1)*

193 LOAD R3 1 *;$temp = 1*

194 STOR R3 [GB +state + 0] *;storeData($temp, 'state', 0)*

195 *;unset($temp)*

196 LOAD R2 0 *;$sleep = 0*

197 BRA calibrateSorter *;calibrateSorter()*

198

199 calibrateSorter: BRS timerManage *;timerManage()*

200 CMP R2 TIMEMOTORDOWN *;if ($sleep == TIMEMOTORDOWN) {*

201 BEQ conditional1

202 return1: ADD R2 1 *;$sleep+=1*

203 BRA calibrateSorter *;calibrateSorter()*

204

205 *;if ($sleep == TIMEMOTORDOWN) {*

206 conditional1: LOAD R3 0 *;$temp = 0*

207 STOR R3 [GB +outputs + HBRIDGE1] *;storeData($temp, 'outputs', HBRIDGE1)*

208 LOAD R4 2 *;$state = 2*

209 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

210 *;unset($state)*

211 LOAD R2 0 *;$sleep = 0*

212 BRA resting *;resting()*

213

214 resting: BRS timerManage *;timerManage()*

215 PUSH R3 *;$startStop = getButtonPressed(0)*

216 LOAD R3 0

217 BRS \_pressed

218 PULL R3

219 SUB SP 5

220 PULL R4

221 ADD SP 4

222 CMP R4 1 *;if ($startStop == 1) {*

223 BEQ conditional2

224 return2: *;unset($startStop)*

225 BRA resting *;resting()*

226

227 *;if ($startStop == 1) {*

228 conditional2: LOAD R3 12 *;$temp = 12*

229 STOR R3 [GB +outputs + LENSLAMPPOSITION] *;storeData($temp, 'outputs', LENSLAMPPOSITION)*

230 *;unset($temp)*

231 BRS timerManage *;timerManage()*

232 PUSH R5 *;sleep(1000)*

233 LOAD R5 1000

234 BRS \_timer

235 PULL R5

236 LOAD R3 12 *;$temp = 12*

237 STOR R3 [GB +outputs + LENSLAMPSORTER] *;storeData($temp, 'outputs', LENSLAMPSORTER)*

238 *;unset($temp)*

239 BRS timerManage *;timerManage()*

240 PUSH R5 *;sleep(2000)*

241 LOAD R5 2000

242 BRS \_timer

243 PULL R5

244 LOAD R3 9 *;$temp = 9*

245 STOR R3 [GB +outputs + CONVEYORBELT] *;storeData($temp, 'outputs', CONVEYORBELT)*

246 LOAD R3 9 *;$temp = 9*

247 STOR R3 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

248 *;unset($temp)*

249 PUSH R5 *;reset timer ;setCountdown(COUNTDOWN)*

250 PUSH R4

251 LOAD R5 -16

252 LOAD R4 0

253 SUB R4 [R5+13]

254 STOR R4 [R5+13] *;set timer*

255 LOAD R4 COUNTDOWN

256 STOR R4 [R5+13]

257 PULL R4

258 PULL R5

259 SETI 8 *;startCountdown()*

260 LOAD R3 3 *;$state = 3*

261 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

262 *;unset($state)*

263 BRA running *;running()*

264

265 running: BRS timerManage *;timerManage()*

266 PUSH R3 *;$startStop = getButtonPressed(0)*

267 LOAD R3 0

268 BRS \_pressed

269 PULL R3

270 SUB SP 5

271 PULL R3

272 ADD SP 4

273 CMP R3 1 *;if ($startStop == 1) {*

274 BEQ conditional3

275 return3: *;unset($startStop)*

276 PUSH R3 *;$position = getButtonPressed(7)*

277 LOAD R3 7

278 BRS \_pressed

279 PULL R3

280 SUB SP 5

281 PULL R3

282 ADD SP 4

283 CMP R3 1 *;if ($position == 1) {*

284 BEQ conditional4

285 return4: *;unset($position)*

286 BRA running *;running()*

287

288 *;if ($startStop == 1) {*

289 conditional3: LOAD R4 0 *;$temp = 0*

290 STOR R4 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

291 *;unset($temp)*

292 PUSH R5 *;reset timer ;setCountdown(BELT \* 10)*

293 PUSH R4

294 LOAD R5 -16

295 LOAD R4 0

296 SUB R4 [R5+13]

297 STOR R4 [R5+13] *;set timer*

298 LOAD R4 BELT \* 10

299 STOR R4 [R5+13]

300 PULL R4

301 PULL R5

302 LOAD R4 9 *;$state = 9*

303 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

304 *;unset($state)*

305 BRA runningTimer *;runningTimer()*

306

307 *;if ($position == 1) {*

308 conditional4: PUSH R5 *;reset timer ;setCountdown(COUNTDOWN)*

309 PUSH R4

310 LOAD R5 -16

311 LOAD R4 0

312 SUB R4 [R5+13]

313 STOR R4 [R5+13] *;set timer*

314 LOAD R4 COUNTDOWN

315 STOR R4 [R5+13]

316 PULL R4

317 PULL R5

318 LOAD R4 4 *;$state = 4*

319 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

320 *;unset($state)*

321 BRA runningWait *;runningWait()*

322

323 runningWait: BRS timerManage *;timerManage()*

324 PUSH R3 *;$startStop = getButtonPressed(0)*

325 LOAD R3 0

326 BRS \_pressed

327 PULL R3

328 SUB SP 5

329 PULL R3

330 ADD SP 4

331 CMP R3 1 *;if ($startStop == 1) {*

332 BEQ conditional5

333 return5: *;unset($startStop)*

334 PUSH R3 *;$position = getButtonPressed(7)*

335 LOAD R3 7

336 BRS \_pressed

337 PULL R3

338 SUB SP 5

339 PULL R3

340 ADD SP 4

341 CMP R3 0 *;if ($position == 0) {*

342 BEQ conditional6

343 return6: *;unset($position)*

344 PUSH R3 *;$colour = getButtonPressed(6)*

345 LOAD R3 6

346 BRS \_pressed

347 PULL R3

348 SUB SP 5

349 PULL R3

350 ADD SP 4

351 CMP R3 1 *;if ($colour == 1) {*

352 BEQ conditional7

353 return7: *;unset($colour)*

354 BRA runningWait *;runningWait()*

355

356 *;if ($startStop == 1) {*

357 conditional5: LOAD R4 0 *;$temp = 0*

358 STOR R4 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

359 *;unset($temp)*

360 PUSH R5 *;reset timer ;setCountdown(BELT \* 10)*

361 PUSH R4

362 LOAD R5 -16

363 LOAD R4 0

364 SUB R4 [R5+13]

365 STOR R4 [R5+13] *;set timer*

366 LOAD R4 BELT \* 10

367 STOR R4 [R5+13]

368 PULL R4

369 PULL R5

370 LOAD R4 9 *;$state = 9*

371 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

372 *;unset($state)*

373 BRA runningTimer *;runningTimer()*

374

375 *;if ($position == 0) {*

376 conditional6: PUSH R5 *;reset timer ;setCountdown(COUNTDOWN)*

377 PUSH R4

378 LOAD R5 -16

379 LOAD R4 0

380 SUB R4 [R5+13]

381 STOR R4 [R5+13] *;set timer*

382 LOAD R4 COUNTDOWN

383 STOR R4 [R5+13]

384 PULL R4

385 PULL R5

386 LOAD R4 5 *;$state = 5*

387 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

388 *;unset($state)*

389 BRA runningTimerReset *;runningTimerReset()*

390

391 *;if ($colour == 1) {*

392 conditional7: LOAD R4 10 *;$temp = 10*

393 STOR R4 [GB +outputs + HBRIDGE0] *;storeData($temp, 'outputs', HBRIDGE0)*

394 LOAD R4 0 *;$temp = 0*

395 STOR R4 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

396 *;unset($temp)*

397 LOAD R4 6 *;$state = 6*

398 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

399 *;unset($state)*

400 BRA motorUp *;motorUp()*

401

402 motorUp: BRS timerManage *;timerManage()*

403 PUSH R3 *;$startStop = getButtonPressed(0)*

404 LOAD R3 0

405 BRS \_pressed

406 PULL R3

407 SUB SP 5

408 PULL R3

409 ADD SP 4

410 CMP R3 1 *;if ($startStop == 1) {*

411 BEQ conditional8

412 return8: *;unset($startStop)*

413 PUSH R3 *;$push = getButtonPressed(5)*

414 LOAD R3 5

415 BRS \_pressed

416 PULL R3

417 SUB SP 5

418 PULL R3

419 ADD SP 4

420 CMP R3 1 *;if ($push == 1) {*

421 BEQ conditional9

422 return9: *;unset($push)*

423 BRA motorUp *;motorUp()*

424

425 *;if ($startStop == 1) {*

426 conditional8: LOAD R4 0 *;$temp = 0*

427 STOR R4 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

428 *;unset($temp)*

429 PUSH R5 *;reset timer ;setCountdown(BELT \* 10)*

430 PUSH R4

431 LOAD R5 -16

432 LOAD R4 0

433 SUB R4 [R5+13]

434 STOR R4 [R5+13] *;set timer*

435 LOAD R4 BELT \* 10

436 STOR R4 [R5+13]

437 PULL R4

438 PULL R5

439 LOAD R4 10 *;$state = 10*

440 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

441 *;unset($state)*

442 BRA motorUpTimer *;motorUpTimer()*

443

444 *;if ($push == 1) {*

445 conditional9: LOAD R4 0 *;$temp = 0*

446 STOR R4 [GB +outputs + HBRIDGE0] *;storeData($temp, 'outputs', HBRIDGE0)*

447 *;unset($temp)*

448 LOAD R4 7 *;$state = 7*

449 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

450 *;unset($state)*

451 LOAD R2 0 *;$sleep = 0*

452 BRA whiteWait *;whiteWait()*

453

454 whiteWait: BRS timerManage *;timerManage()*

455 CMP R2 SORT *;if ($sleep == SORT) {*

456 BEQ conditional10

457 return10: PUSH R3 *;$startStop = getButtonPressed(0)*

458 LOAD R3 0

459 BRS \_pressed

460 PULL R3

461 SUB SP 5

462 PULL R3

463 ADD SP 4

464 CMP R3 1 *;if ($startStop == 1) {*

465 BEQ conditional11

466 return11: *;unset($startStop)*

467 ADD R2 1 *;$sleep+=1*

468 BRA whiteWait *;whiteWait()*

469

470 *;if ($sleep == SORT) {*

471 conditional10: LOAD R3 10 *;$temp = 10*

472 STOR R3 [GB +outputs + HBRIDGE1] *;storeData($temp, 'outputs', HBRIDGE1)*

473 *;unset($temp)*

474 PUSH R5 *;reset timer ;setCountdown(COUNTDOWN)*

475 PUSH R4

476 LOAD R5 -16

477 LOAD R4 0

478 SUB R4 [R5+13]

479 STOR R4 [R5+13] *;set timer*

480 LOAD R4 COUNTDOWN

481 STOR R4 [R5+13]

482 PULL R4

483 PULL R5

484 LOAD R3 8 *;$state = 8*

485 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

486 *;unset($state)*

487 LOAD R2 0 *;$sleep = 0*

488 BRA motorDown *;motorDown()*

489

490 *;if ($startStop == 1) {*

491 conditional11: LOAD R4 0 *;$temp = 0*

492 STOR R4 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

493 *;unset($temp)*

494 PUSH R5 *;reset timer ;setCountdown(BELT \* 10)*

495 PUSH R4

496 LOAD R5 -16

497 LOAD R4 0

498 SUB R4 [R5+13]

499 STOR R4 [R5+13] *;set timer*

500 LOAD R4 BELT \* 10

501 STOR R4 [R5+13]

502 PULL R4

503 PULL R5

504 LOAD R4 11 *;$state = 11*

505 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

506 *;unset($state)*

507 BRA whiteWaitTimer *;whiteWaitTimer()*

508

509 whiteWaitTimer: BRS timerManage *;timerManage()*

510 LOAD R3 15 *;$state = 15*

511 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

512 *;unset($state)*

513 BRA whiteWaitStop *;whiteWaitStop()*

514

515 whiteWaitStop: BRS timerManage *;timerManage()*

516 CMP R2 SORT *;if ($sleep == SORT) {*

517 BEQ conditional12

518 return12: ADD R2 1 *;$sleep+=1*

519 BRA whiteWaitStop *;whiteWaitStop()*

520

521 *;if ($sleep == SORT) {*

522 conditional12: LOAD R3 10 *;$temp = 10*

523 STOR R3 [GB +outputs + HBRIDGE1] *;storeData($temp, 'outputs', HBRIDGE1)*

524 LOAD R3 0 *;$temp = 0*

525 STOR R3 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

526 *;unset($temp)*

527 LOAD R3 12 *;$state = 12*

528 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

529 *;unset($state)*

530 LOAD R2 0 *;$sleep = 0*

531 BRA motorDownStop *;motorDownStop()*

532

533 motorDownStop: BRS timerManage *;timerManage()*

534 CMP R2 TIMEMOTORDOWN *;if ($sleep == TIMEMOTORDOWN) {*

535 BEQ conditional13

536 return13: ADD R2 1 *;$sleep+=1*

537 BRA motorDownStop *;motorDownStop()*

538

539 *;if ($sleep == TIMEMOTORDOWN) {*

540 conditional13: LOAD R3 0 *;$temp = 0*

541 STOR R3 [GB +outputs + HBRIDGE1] *;storeData($temp, 'outputs', HBRIDGE1)*

542 *;unset($temp)*

543 LOAD R3 9 *;$state = 9*

544 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

545 *;unset($state)*

546 LOAD R2 0 *;$sleep = 0*

547 BRA runningStop *;runningStop()*

548

549 runningStop: BRS timerManage *;timerManage()*

550 PUSH R3 *;$colour = getButtonPressed(6)*

551 LOAD R3 6

552 BRS \_pressed

553 PULL R3

554 SUB SP 5

555 PULL R3

556 ADD SP 4

557 CMP R3 1 *;if ($colour == 1) {*

558 BEQ conditional14

559 return14: *;unset($colour)*

560 BRA runningStop *;runningStop()*

561

562 *;if ($colour == 1) {*

563 conditional14: LOAD R4 10 *;$temp = 10*

564 STOR R4 [GB +outputs + HBRIDGE0] *;storeData($temp, 'outputs', HBRIDGE0)*

565 LOAD R4 0 *;$temp = 0*

566 STOR R4 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

567 *;unset($temp)*

568 LOAD R4 10 *;$state = 10*

569 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

570 *;unset($state)*

571 BRA motorUpStop *;motorUpStop()*

572

573 motorUpStop: BRS timerManage *;timerManage()*

574 PUSH R3 *;$push = getButtonPressed(5)*

575 LOAD R3 5

576 BRS \_pressed

577 PULL R3

578 SUB SP 5

579 PULL R3

580 ADD SP 4

581 CMP R3 1 *;if ($push == 1) {*

582 BEQ conditional15

583 return15: *;unset($push)*

584 BRA motorUpStop *;motorUpStop()*

585

586 *;if ($push == 1) {*

587 conditional15: LOAD R4 0 *;$temp = 0*

588 STOR R4 [GB +outputs + HBRIDGE0] *;storeData($temp, 'outputs', HBRIDGE0)*

589 *;unset($temp)*

590 LOAD R4 11 *;$state = 11*

591 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

592 *;unset($state)*

593 BRA whiteWaitStop *;whiteWaitStop()*

594

595 motorDown: BRS timerManage *;timerManage()*

596 PUSH R3 *;$colour = getButtonPressed(6)*

597 LOAD R3 6

598 BRS \_pressed

599 PULL R3

600 SUB SP 5

601 PULL R3

602 ADD SP 4

603 CMP R3 1 *;if ($colour == 1) {*

604 BEQ conditional16

605 return16: *;unset($colour)*

606 CMP R2 TIMEMOTORDOWN *;if ($sleep == TIMEMOTORDOWN) {*

607 BEQ conditional17

608 return17: PUSH R3 *;$startStop = getButtonPressed(0)*

609 LOAD R3 0

610 BRS \_pressed

611 PULL R3

612 SUB SP 5

613 PULL R3

614 ADD SP 4

615 CMP R3 1 *;if ($startStop == 1) {*

616 BEQ conditional18

617 return18: *;unset($startStop)*

618 ADD R2 1 *;$sleep+=1*

619 BRA motorDown *;motorDown()*

620

621 *;if ($colour == 1) {*

622 conditional16: LOAD R4 0 *;$temp=0*

623 STOR R4 [GB +outputs + HBRIDGE1] *;storeData($temp,'outputs',HBRIDGE1)*

624 LOAD R4 10 *;$temp = 10*

625 STOR R4 [GB +outputs + HBRIDGE0] *;storeData($temp, 'outputs', HBRIDGE0)*

626 *;unset($temp)*

627 LOAD R4 6 *;$state = 6*

628 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

629 LOAD R2 0 *;$sleep=0*

630 *;unset($state)*

631 BRA motorUp *;motorUp()*

632

633 *;if ($sleep == TIMEMOTORDOWN) {*

634 conditional17: LOAD R3 0 *;$temp = 0*

635 STOR R3 [GB +outputs + HBRIDGE1] *;storeData($temp, 'outputs', HBRIDGE1)*

636 LOAD R3 7 *;$temp = 7*

637 STOR R3 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

638 *;unset($temp)*

639 LOAD R3 4 *;$state = 4*

640 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

641 LOAD R2 0 *;$sleep = 0*

642 *;unset($state)*

643 BRA runningWait *;runningWait()*

644

645 *;if ($startStop == 1) {*

646 conditional18: LOAD R4 0 *;$temp = 0*

647 STOR R4 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

648 *;unset($temp)*

649 PUSH R5 *;reset timer ;setCountdown(BELT \* 10)*

650 PUSH R4

651 LOAD R5 -16

652 LOAD R4 0

653 SUB R4 [R5+13]

654 STOR R4 [R5+13] *;set timer*

655 LOAD R4 BELT \* 10

656 STOR R4 [R5+13]

657 PULL R4

658 PULL R5

659 LOAD R4 12 *;$state = 12*

660 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

661 *;unset($state)*

662 BRA motorDownTimer *;motorDownTimer()*

663

664 motorDownTimer: BRS timerManage *;timerManage()*

665 LOAD R3 16 *;$state = 16*

666 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

667 *;unset($state)*

668 BRA motorDownStop *;motorDownStop()*

669

670 motorUpTimer: BRS timerManage *;timerManage()*

671 LOAD R3 14 *;$state = 14*

672 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

673 *;unset($state)*

674 BRA motorUpStop *;motorUpStop()*

675

676 runningTimerReset: BRS timerManage *;timerManage()*

677 LOAD R3 4 *;$state = 4*

678 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

679 *;unset($state)*

680 BRA runningWait *;runningWait()*

681

682 runningTimer: BRS timerManage *;timerManage()*

683 LOAD R3 13 *;$state = 13*

684 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

685 *;unset($state)*

686 BRA runningStop *;runningStop()*

687

688 *;if ($location == 0) {*

689 conditional19: LOAD R3 0 *;$engines = 0*

690 BRA return19 *;}*

691

692 *;if ($voltage > $counter) {*

693 conditional20: LOAD R4 R1 *;$voltage = $location*

694 PUSH R5 *;$voltage = pow(2, $voltage)*

695 LOAD R5 2

696 BRS \_pow

697 LOAD R4 R5

698 PULL R5

699 ADD R3 R4 *;$engines += $voltage*

700 BRA return20 *;}*

701

702 *;if ($location == 7) {*

703 conditional21: PUSH R5 *;sleep(1)*

704 LOAD R5 1

705 BRS \_timer

706 PULL R5

707 PUSH R5 *;display($engines, 'leds')*

708 LOAD R5 -16

709 STOR R3 [R5+11]

710 PULL R5

711 *;unset($voltage)*

712 PUSH R3 *;$abort = getButtonPressed(1)*

713 LOAD R3 1

714 BRS \_pressed

715 PULL R3

716 SUB SP 5

717 PULL R4

718 ADD SP 4

719 CMP R4 1 *;if ($abort == 1) {*

720 BEQ conditional22

721 return22: *;unset($abort)*

722 CMP R0 6 *;if ($counter == 6) {*

723 BEQ conditional23

724 return23: CMP R0 11 *;if ($counter == 11) {*

725 BEQ conditional24

726 return24: LOAD R3 0 *;$engines = 0*

727 LOAD R1 0 *;$location = 0*

728 ADD R0 1 *;$counter+=1*

729 RTS *;return*

730 BRA return21 *;}*

731

732 *;if ($abort == 1) {*

733 conditional22: BRA abort *;abort()*

734 BRA return22 *;}*

735

736 *;if ($counter == 6) {*

737 conditional23: LOAD R4 [ GB + state + 0 ] *;$temp = getData('state', 0)*

738 MOD R4 10 *;mod(10, $temp)*

739 PUSH R5 *;display($temp, 'display', 1)*

740 PUSH R4

741 LOAD R5 R4

742 BRS \_Hex7Seg

743 LOAD R4 %0000001

744 STOR R4 [R5+9]

745 PULL R4

746 PULL R5

747 *;unset($temp)*

748 BRA return23 *;}*

749

750 *;if ($counter == 11) {*

751 conditional24: PUSH R2 *;pushStack($sleep)*

752 LOAD R4 [ GB + state + 0 ] *;$temp = getData('state', 0)*

753 LOAD R2 R4 *;$sleep = $temp*

754 MOD R2 10 *;mod(10, $sleep)*

755 SUB R4 R2 *;$temp -= $sleep*

756 DIV R4 10 *;$temp /= 10*

757 PUSH R5 *;display($temp, 'display', 2)*

758 PUSH R4

759 LOAD R5 R4

760 BRS \_Hex7Seg

761 LOAD R4 %0000010

762 STOR R4 [R5+9]

763 PULL R4

764 PULL R5

765 PULL R2 *;pullStack($sleep)*

766 *;unset($temp)*

767 BRA return24 *;}*

768

769 abort: *;unset($engines)*

770 PUSH R5 *;reset timer ;setCountdown(1000)*

771 PUSH R4

772 LOAD R5 -16

773 LOAD R4 0

774 SUB R4 [R5+13]

775 STOR R4 [R5+13] *;set timer*

776 LOAD R4 1000

777 STOR R4 [R5+13]

778 PULL R4

779 PULL R5

780 LOAD R3 [ GB + stackPointer + 0 ] *;$temp = getData('stackPointer', 0)*

781 LOAD SP R3 *;setStackPointer($temp)*

782 LOAD R3 0 *;$temp = 0*

783 STOR R3 [GB +outputs + HBRIDGE1] *;storeData($temp, 'outputs', HBRIDGE1)*

784 STOR R3 [GB +outputs + HBRIDGE0] *;storeData($temp, 'outputs', HBRIDGE0)*

785 STOR R3 [GB +outputs + LENSLAMPPOSITION] *;storeData($temp, 'outputs', LENSLAMPPOSITION)*

786 STOR R3 [GB +outputs + LENSLAMPSORTER] *;storeData($temp, 'outputs', LENSLAMPSORTER)*

787 STOR R3 [GB +outputs + LEDSTATEINDICATOR] *;storeData($temp, 'outputs', LEDSTATEINDICATOR)*

788 STOR R3 [GB +outputs + DISPLAY] *;storeData($temp, 'outputs', DISPLAY)*

789 STOR R3 [GB +outputs + CONVEYORBELT] *;storeData($temp, 'outputs', CONVEYORBELT)*

790 STOR R3 [GB +outputs + FEEDERENGINE] *;storeData($temp, 'outputs', FEEDERENGINE)*

791 *;unset($temp)*

792 BRS timerManage *;timerManage()*

793 LOAD R3 17 *;$state = 17*

794 STOR R3 [GB +state + 0] *;storeData($state, 'state', 0)*

795 LOAD R3 7 *;$state = 7*

796 PUSH R5 *;display($state, 'leds2', 0)*

797 LOAD R5 -16

798 STOR R3 [R5+10]

799 PULL R5

800 *;unset($state)*

801 BRA aborted *;aborted()*

802

803 aborted: PUSH R5 *;reset timer ;setCountdown(1000)*

804 PUSH R4

805 LOAD R5 -16

806 LOAD R4 0

807 SUB R4 [R5+13]

808 STOR R4 [R5+13] *;set timer*

809 LOAD R4 1000

810 STOR R4 [R5+13]

811 PULL R4

812 PULL R5

813 BRS timerManage *;timerManage()*

814 PUSH R3 *;$startStop = getButtonPressed(0)*

815 LOAD R3 0

816 BRS \_pressed

817 PULL R3

818 SUB SP 5

819 PULL R3

820 ADD SP 4

821 CMP R3 1 *;if ($startStop == 1) {*

822 BEQ conditional25

823 return25: *;unset($startStop)*

824 BRA aborted *;aborted()*

825

826 *;if ($startStop == 1) {*

827 conditional25: LOAD R4 10 *;$temp = 10*

828 STOR R4 [GB +outputs + HBRIDGE0] *;storeData($temp, 'outputs', HBRIDGE0)*

829 *;unset($temp)*

830 LOAD R4 0 *;$state = 0*

831 STOR R4 [GB +state + 0] *;storeData($state, 'state', 0)*

832 *;unset($state)*

833 BRA initial *;initial()*

834

835 timerManage: CMP R1 0 *;if ($location == 0) {*

836 BEQ conditional19

837 return19: MOD R0 12 *;mod(12, $counter)*

838 ADD R1 outputs *;$voltage = getData('outputs', $location)*

839 LOAD R4 [ GB + R1]

840 SUB R1 outputs

841 CMP R4 R0 *;if ($voltage > $counter) {*

842 BGT conditional20

843 return20: CMP R1 7 *;if ($location == 7) {*

844 BEQ conditional21

845 return21: ADD R1 1 *;$location+=1*

846 BRA timerManage *;branch('timerManage')*

847

848 @END