Photovoltaic Martian Bugs

Project Proposal

Adam Jackson & Matt Travis

Project Advisors
Dr. Huggins & Dr. Malinowski

Project Summary

The Photovoltaic Martian Bug project strives to create small autonomous robots that are powered by onboard photovoltaic cells. The bugs will be driven by two electric motors that will allow it to maneuver with a zero turning radius. These motors will be controlled with an 8051-type microcontroller that will interpret the sensory inputs and move the bug accordingly. The microcontroller will accept inputs from contact sensors, light sensors, and an infrared transceiver to determine operational mode. These modes will allow the bug to intelligently react to its environment and other bugs.

Functional Description

The Photovoltaic Martian Bugs are small, autonomous, mobile, solar-powered robots equipped with a basic sensor array. The inputs will include one or more collision detectors, several buttons, two light meters, and an infrared transceiver. These inputs control the operation of the bugs in the primary operating modes. The outputs include status LEDs, infrared communication, and motion. The inputs and outputs are depicted in Figure 1.

The bugs have two modes of operation as shown in Figure 2. In the default sleep mode, the bugs wait for the light level detected by the light meter to be sufficient to enter active mode. Once in active mode the bug moves and interacts with its environment according to one or more behavioral settings. The backup battery will charge during this mode if sufficient power is available. If the light meter detects that the light level has become insufficient to maintain active mode, the controller circuitry will be alerted, and the bug will gracefully return to sleep mode.

The bugs have four behaviors during active mode. In the default random walk behavior, the bug moves N units forward, and then turns M degrees, continuously, where N and M are randomly chosen at each step. While the bug is moving it may make contact with obstacles (walls, feet, other bugs, etc.). In such an event the bug will stop, back up a fixed distance and turn 45 degrees away from the object. The bug will determine what direction to turn with a set of antenna-like feelers attached to contact switches. When the bug makes contact with one feeler it will back up and turn away from the object. For example, if the bug were to make contact with the left feeler, it would back up and turn right. If it made contact with both feelers at the same time it would back up and randomly turn left or right.

The bugs will remain in random walk mode until the light level drops below satisfactory levels. If the light level becomes too low to maintain continuous operation, the bug will enter into the forage behavior and begin moving towards the area with more light. It will continue foraging until the light level returns to a predefined acceptable range or a sufficient energy source is located.

If the bug encounters a particularly well-lit area, it will stop moving and transmit an infrared signal indicating the light level in the area. It will continue this behavior for a set period of time to allow other bugs to detect the signal and converge on the well-lit area. After this, the bug will go back to foraging for even stronger light sources.

As suggested above, during the forage mode, the bug may receive a beacon signal from another bug that has found a strong light source. If this happens, the foraging bug will change to convergence mode and move towards the beaconing bug. It will continue to do this either as long as the light level keeps increasing or as long as it keeps receiving beacon signals. Once the light level has stabilized the converging bug will resume foraging for light sources.

During any of these behaviors in the active mode, the bug may receive input from several operator buttons mounted on the bug's body. This is shown in Figure 2. A halt button would transition the bug to a hibernation state, in which the bug remains powered on but does not move. This state can be seen in the mode diagram in Figure 2 as the soft off feature. Such a mode could be used to repair the sensors or locomotion systems of the bug "live", or to attach the bug to a host system for debugging. Pushing the halt button again would return the bug to the previous active behavior. A second button would be a true power button that would disconnect the battery and solar panels from the rest of the system. This state is also depicted in Figure 2 as the hard off feature.

At any time during the active mode, the power may be interrupted. This includes pushing the power button, running the battery out, or insufficient current from the solar cells. For the latter condition, the bug would gracefully enter a low power consumption state, powering off the motors and the active sensors and switching the microcontroller to standby, with all other available current directed to trickle-charging the battery. Once the light level returns the bug would power up normally. The other two conditions are "hard power-off" and "soft power-off". Pushing the power button is hard power-off; the solar cells will continue to charge the battery, but the bug will be incapable of returning to active mode on its own. The battery also charges in the soft power-off condition, but the bug will be able to resume upon proper sensory input.

To provide real time status of the bugs current operational mode and behavioral status a dual color LED will be used. The LED will change color according to the current mode of operation. To determine the behavioral state in each of the modes the LED will be pulsed in a unique pattern for each behavior.

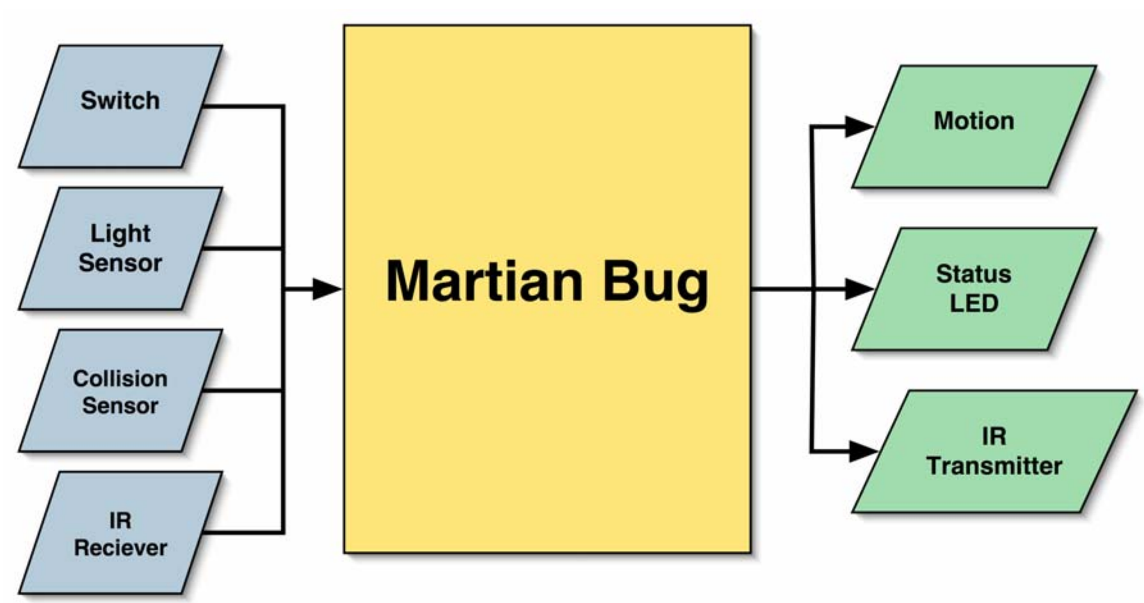


Figure 1: Overall Block Diagram

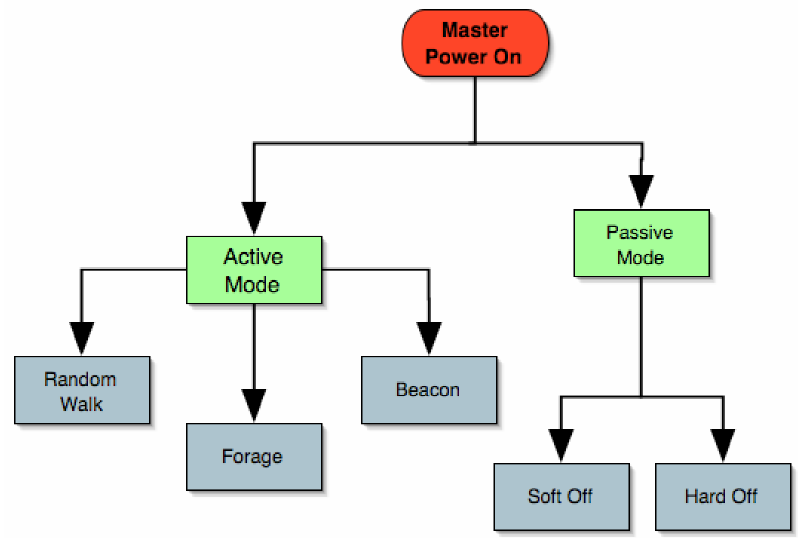


Figure 2: Mode Organizational Chart

System Block Diagram

Hardware Subsystems

The Photovoltaic Martian Bugs have the following hardware subsystems. Each subsystem is listed with its inputs and outputs, along with an operational description for each mode. Refer to Figure 3 for the hardware organization and Figure 4 for the hardware flowchart.

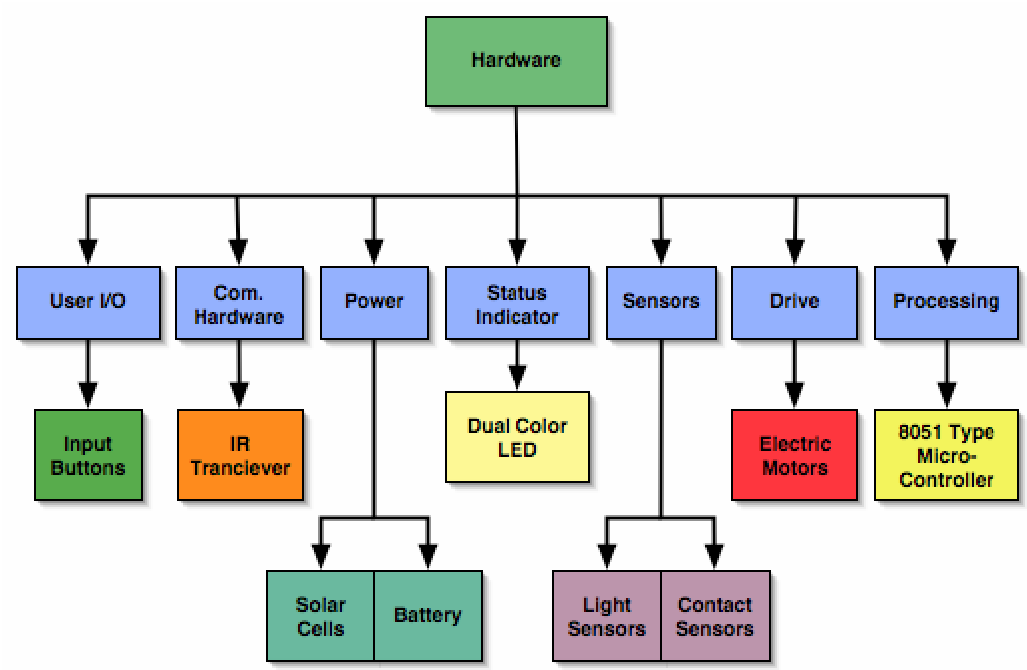


Figure 3: Hardware organizational chart

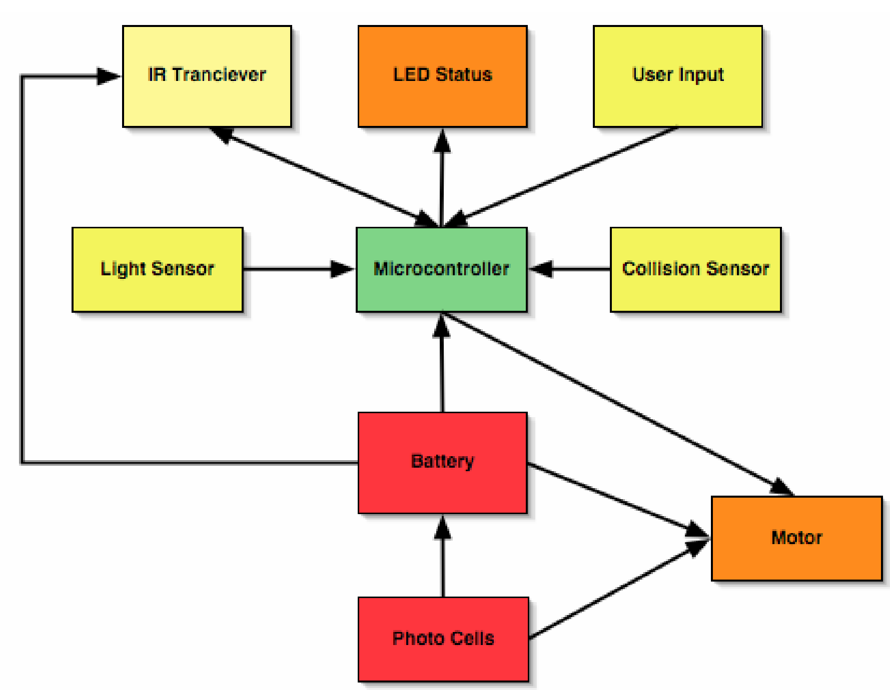


Figure 4: Hardware Flow

Subsystem: Light level sensor
Inputs: Photons
Outputs: Light level signal
Operation:

The bug is equipped with two or more light level sensors. Each of these sensors are mounted in a different position on the bug, each facing a different direction. In all modes, including the "off" modes, these sensors generate a signal indicating the current light level. This signal is used to determine whether the light level is sufficient to continue operation. In Forage mode, the microcontroller compares the light signals from the sensors and walks towards the region with more light. If the light level exceeds some threshold, the bug stops foraging and begins transmitting an infrared signal for other bugs to home in on.

Subsystem: Infrared transceiver
Inputs: Infrared data
Outputs: Infrared data
Operation:

The bug is equipped with two or more infrared transceivers. These are used for communication between bugs. In the Forage mode, the bug listens to the IR transceiver. If it receives a transmission from another bug indicating a good light source, the bug enters Tracking mode. In the Tracking mode, the microcontroller compares the relative strength of the received signals to determine the location of the bug sending the transmissions. To determine location a more than one transceiver is required. If the bug receives a signal on its left receiver it will turn left and move forward. The tracking bug continues to execute these moves and turns to converge on the position of the sending bug. If a signal is detected on more than one receiver the strongest of the signals will be chosen. In Transmission mode, the microcontroller sends out an IR signal on the transceiver to alert other bugs in the area of a strong light source. It waits for some period of time to allow the other bugs to converge on the signal, and then returns to Random-Walk mode. In all other modes, the IR transceivers are powered off.

Subsystem: Photovoltaic Cells
Inputs: Photons
Outputs: Power
Operation:

In all modes, the photovoltaic cells convert photons to electric current. This current is used to charge the backup battery, move the bug, and process the various other input signals the bug receives.

Subsystem: Battery
Inputs: Trickle-charge current
Outputs: Back-up current
Operation:

In all modes, the battery supplies the operational current needed to run the bug. It acts as a voltage source in parallel with the photovoltaic cells. The battery is not the primary power source. It is only intended to supplement the power from the solar cells during high load or low light conditions.

Subsystem: Collision sensors
Inputs: Hard objects
Outputs: Collision signal
Operation:

In all the moving modes (Tracking, Forage, Random-Walk), the collision sensors detect collisions with other objects. The microcontroller uses this signal to correct the path of the bug to avoid the object. In all other modes the collisions sensors have no effect.

Subsystem: Motor and power electronics
Inputs: Motion signal
Outputs: Kinetic energy
Operation:

In all the moving modes (Tracking, Forage, Random-Walk), the microcontroller sends motion signals to the motor to move the bug. The bug is capable of forward and reverse motion and can turn. In all the other modes this subsystem does nothing.

Subsystem: User I/O
Inputs: Button presses
Outputs: Blinking LEDs
Operation:

In all modes, the LEDs blink to indicate the current mode of operation. The user can change the operating mode by pressing one of the buttons (Sleep or Power).

Subsystem: Microcontroller
Inputs: Light level, collision signal, button presses, infrared data
Outputs: LED signals, motion signal, infrared data
Operation:

The microcontroller determines the current mode from the various inputs. Based on the current mode, the microcontroller generates the desired output signals. This is described in more detail in the next section.

Software Subsystem

The following is a breakdown of each of the software subsystems. The overall software flow chart seen in Figure 5 shows the interconnections between the software modules and how the system moves from state to state.

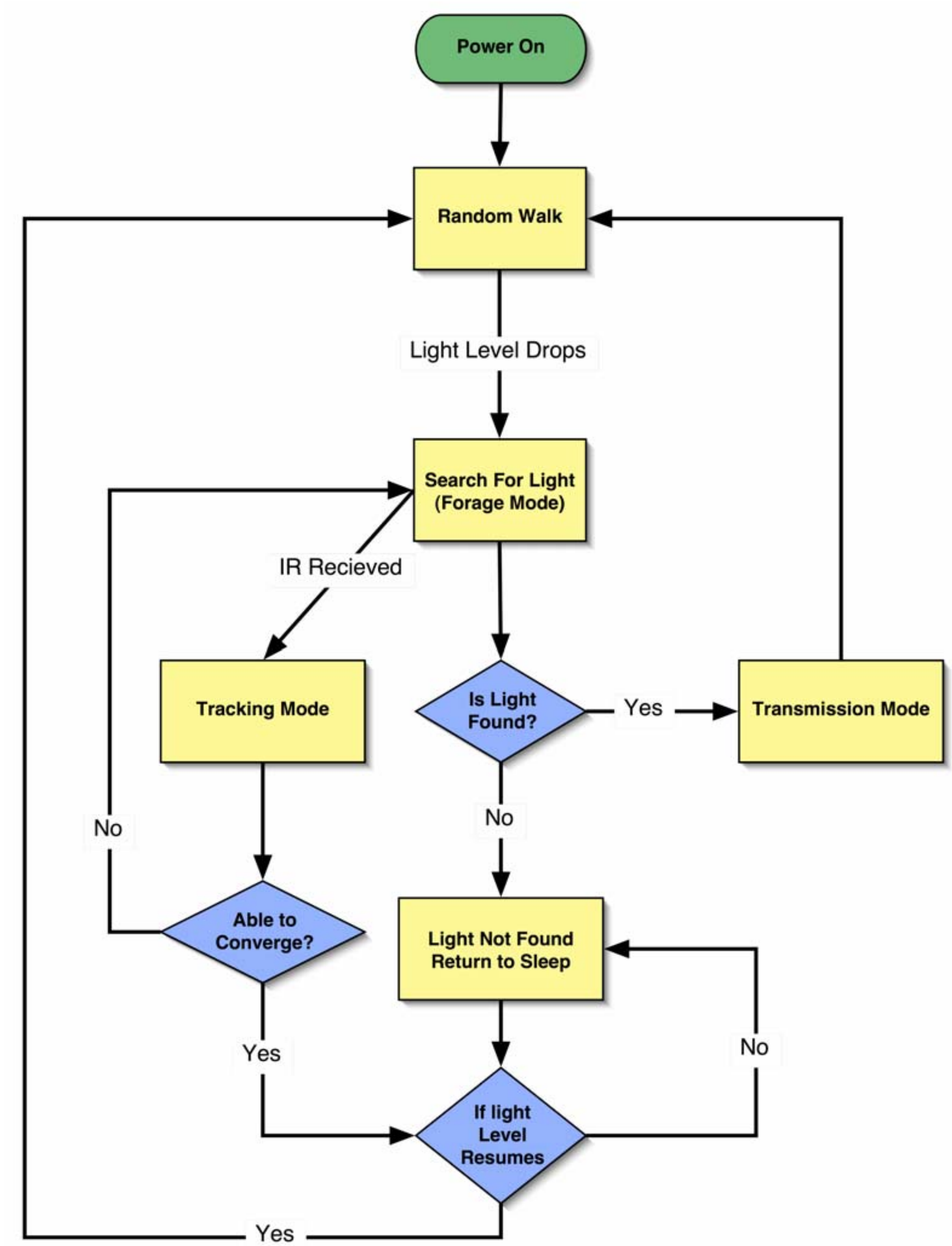


Figure 5: High Level Software Flow

Subsystem: Random Walk Mode
Inputs: Light level data, collision signals, button presses
Outputs: Motion
Operation:

In random walk mode the bug will move a random distance X then turn in a random direction Y. The distances and direction will not be entirely random but will have enough combinations to have the appearance of randomness. This mode will also monitor the light level being input from the light sensors. If the level drops too far it will enter into Forage mode. If the bug encounters an object it will stop, back up, and turn a specified direction away from the object, and continue. Figure 6 seen below shows The anticipated software flow.

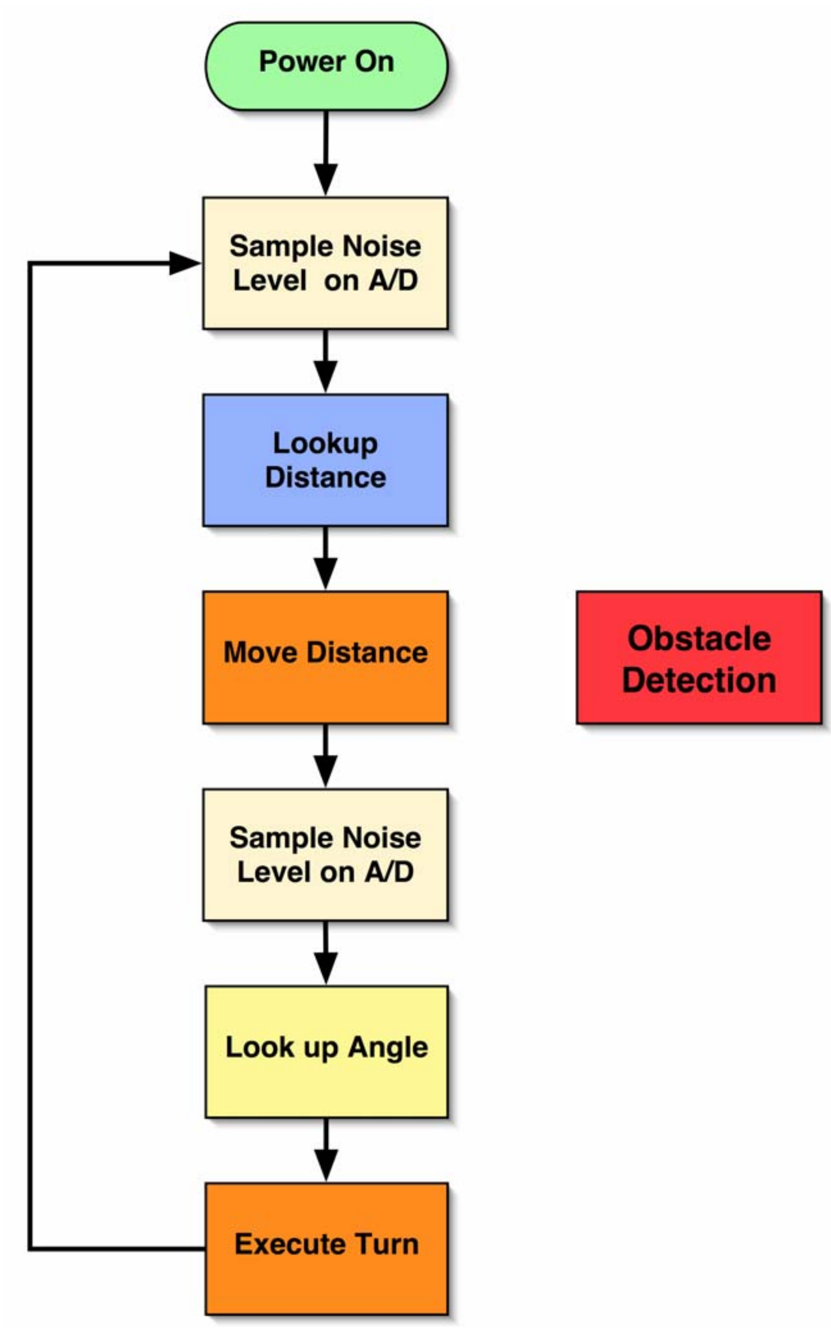


Figure 6: Random Walk Flow Chart

Subsystem: Forage Mode
Inputs: Light level, collision signal, button presses, infrared data
Outputs: Motion
Operation:

In Forage mode the bug attempts to find a stronger light source. In this mode its movements are no longer random but are based upon the inputs of the light sensors. If one light sensor produces a higher light level than the other the bug will turn in the direction of that sensor and continue forward. If the light levels are the same on both sensors the bug will continue to move straight forward. The bug will move in this manner until the light sensors read adequate light levels. This mode maintains the same procedure for encountering objects in the random walk mode. This mode also accepts signals from an infrared receiver. These signals indicate that another bug has found light and the receiving bug will attempt to locate the transmitting bug. If no signals are received and no light can be found after a certain amount of time the bug will enter sleep mode. Figure 7 below shows the proposed forage mode software flow.

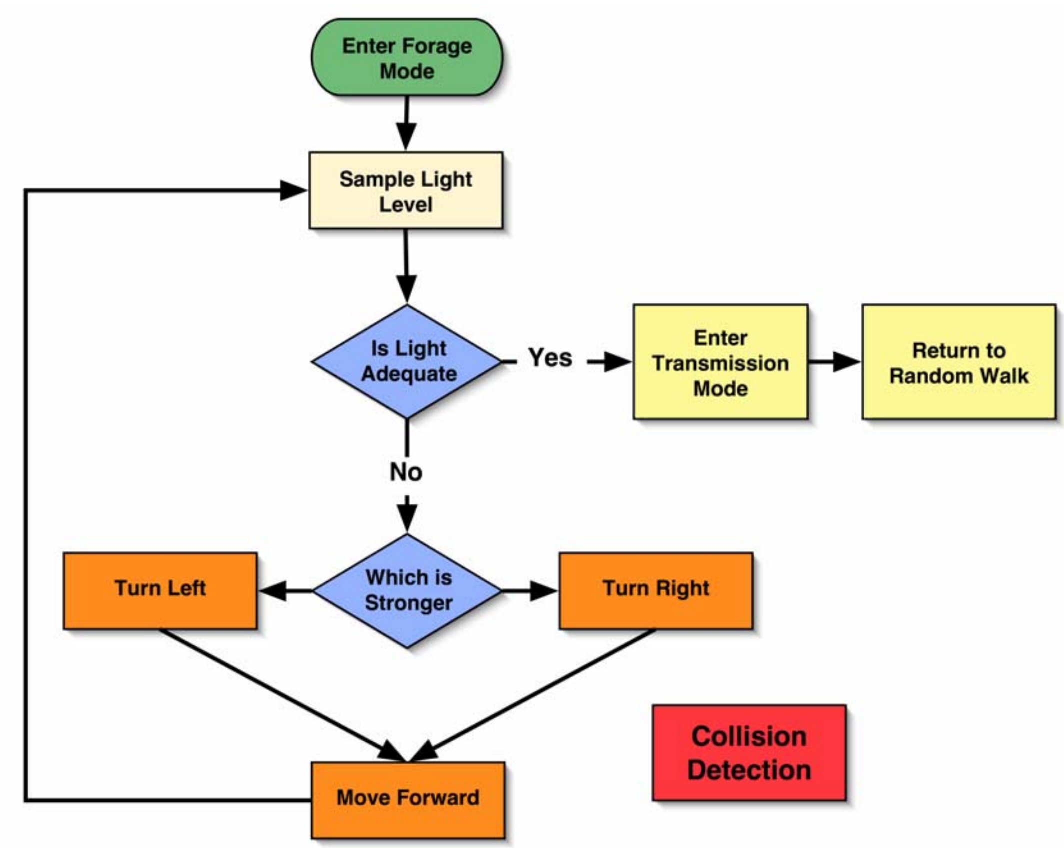


Figure 7: Forage Mode Flow

Subsystem: Tracking Mode
Inputs: Light level, collision signal, button presses, infrared data
Outputs: Motion, status LED
Operation:

If the bug is in Forage mode and looking for light it may receive an infrared signal from a second bug. This signal will tell the bug that there is light in a general direction. The receiving bug will detect which IR receiver has the strongest signal and turn in that direction and continue. The bug will continue doing this until it reaches and adequate light level or it loses the IR signal. If it loses the signal, it will return to forage mode and continue. Figure 8 shows the proposed software flow for the infrared receiver software.

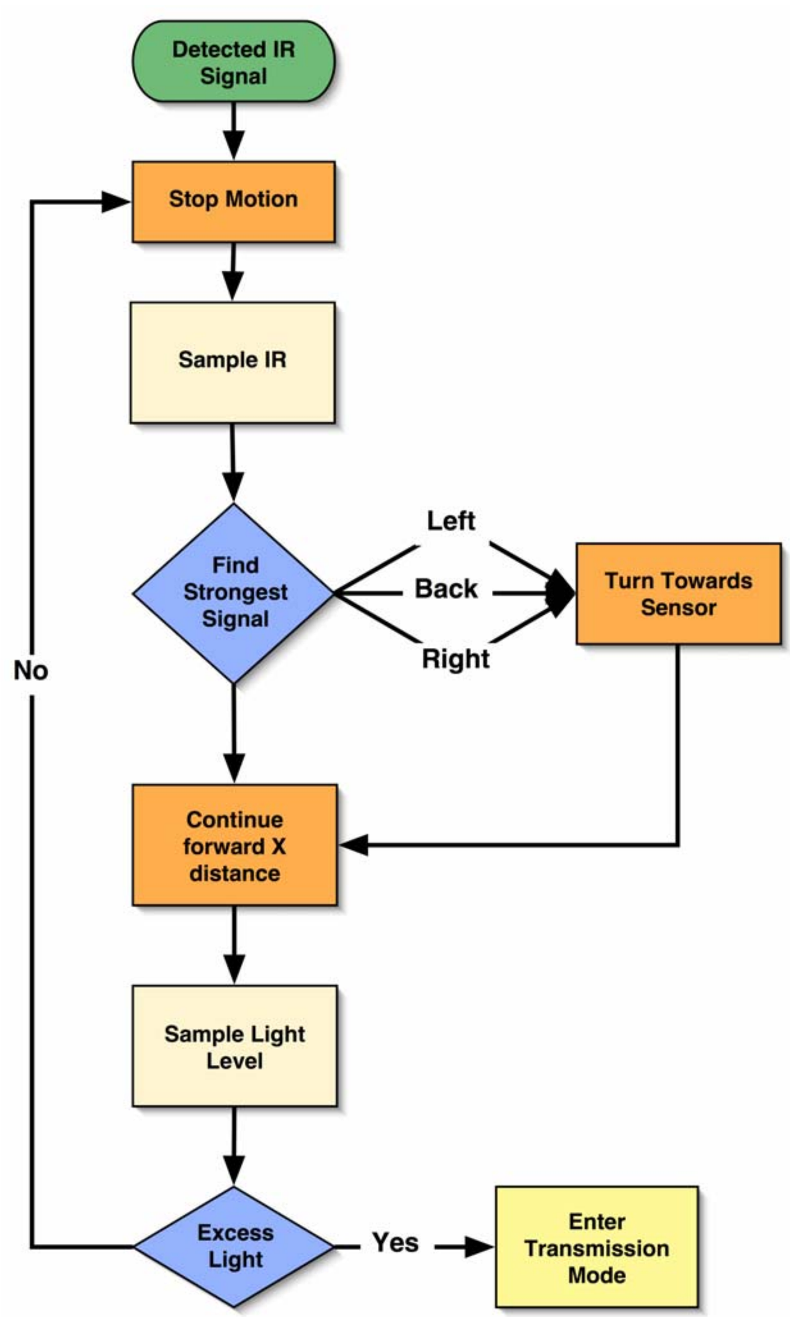


Figure 8: IR Receive Mode

Subsystem: Transmission Mode
Inputs: Light level, button presses
Outputs: LED signals, infrared data
Operation:

When a bug finds a light source that is adequate to exit forage mode it will stop moving and begin to pulse out data on the IR transmitter. These signals indicate to other bugs that there is light at its location. Bugs receiving this signal will then attempt to converge on its location. Figure 9 shown below is the proposed flowchart for the infrared transmission software.

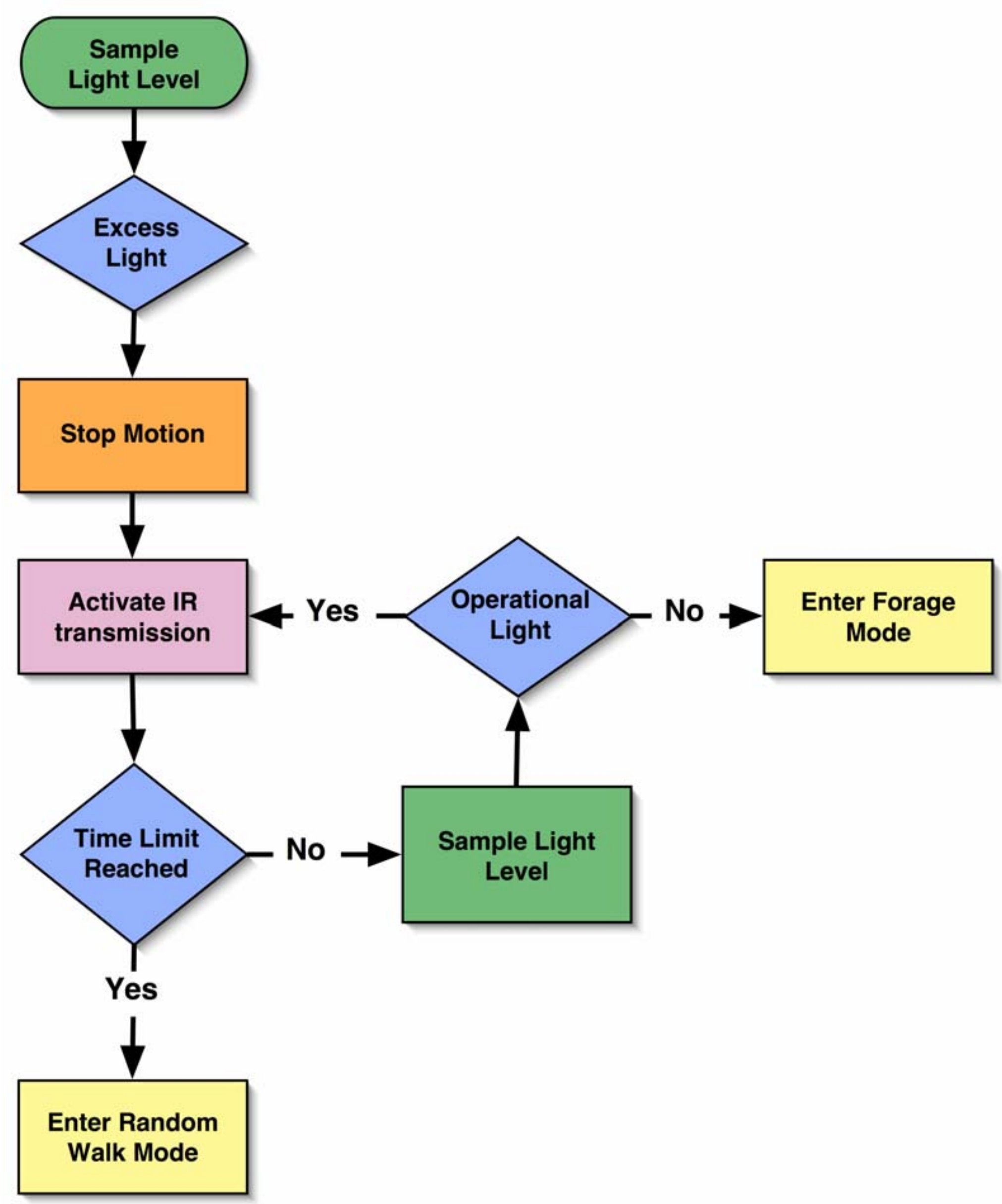


Figure 9: IR Transmission Flow

Subsystem: Collision Mode
Inputs: Contact sensors, button presses
Outputs: LED signals, motion
Operation:

When the bug encounters an object it will stop its forward motion. The bug will then decide whether the obstacle is on its right, left or in front of it. The bug will then back up a set distance and turn away from the object and proceed forward. Figure 10 shown below is the proposed software flow for the collision mode software.

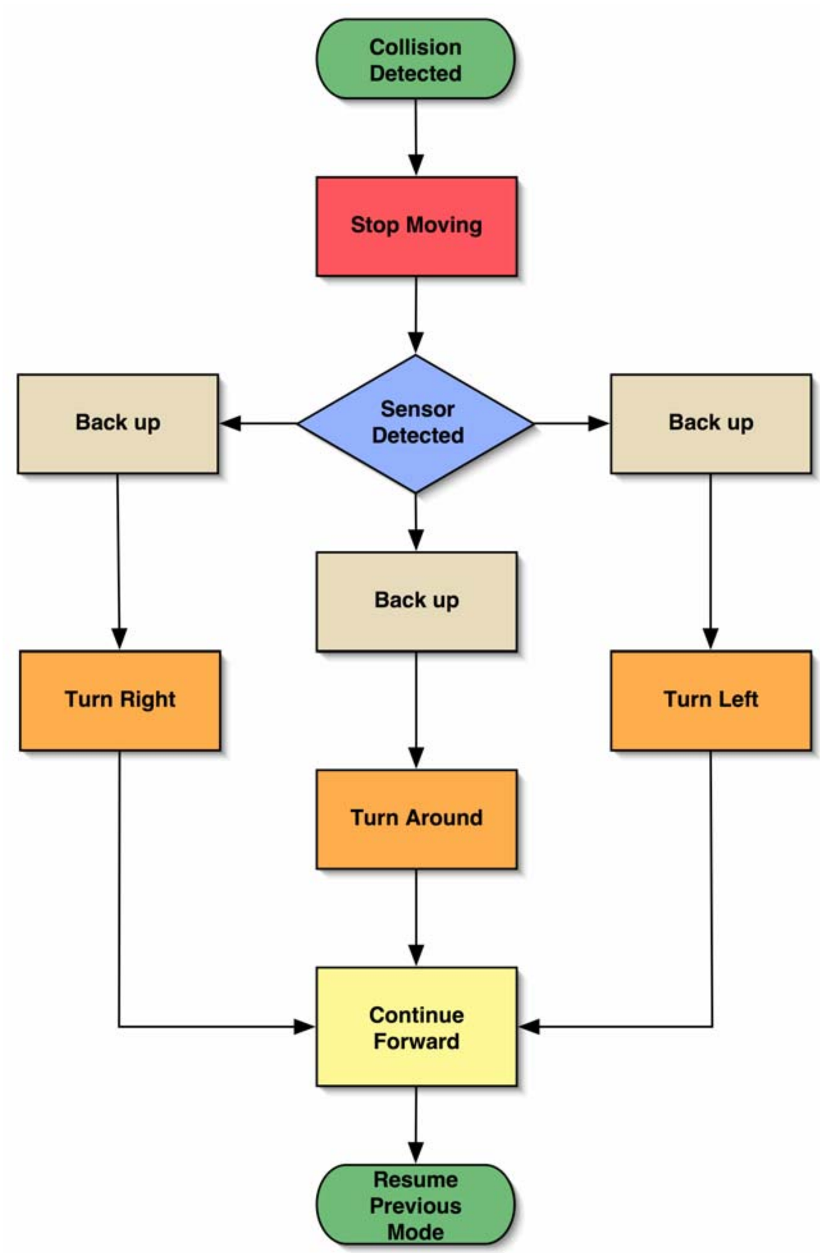


Figure 10: Collision Detection Flow

Subsystem: Sleep Mode
Inputs: Light level, button presses
Outputs: LED signals
Operation:

In sleep mode the bug does nothing besides wait. The bug waits for light to return or for an input from the user. While in this mode the solar cells collect what light they can to charge the battery and run the status LED.

Subsystem: Death Mode
Inputs: Photons, lack of
Outputs: None
Operation:

If the bug is in standby mode too long it has the possibility of depleting all of its energy sources. The system requires a very minimal amount of power in standby but if such power is not available the bug will not be able to continue operation. If this occurs the bug will essentially die. The bug will stay in this mode until human intervention revives it from this mode.

Design Strategy

The problem in developing a solar powered system is that power requirements are relative to other variables. The amount of power needed to move the bug is dependant upon how large the bug is and how fast it is to be moved. The problem can also be viewed in the opposite light starting with the solar power output. The size and weight of the system could be determined if the size of the panels are fixed. Each approach involves making assumptions of key values such as weight, surface area, or cell efficiency.

The single most important variable is mass. The mass of the bug will determine the motor size needed to move the bug. Knowing the weight and motor size we can gauge how much power we will need. We can then take this power requirement and translate it to a size and type of solar panel needed to power the system. The benefit of choosing a mass is that only one assumption is made instead of many. However, this provides a single point of failure. If the guess is to low the bug will be too heavy and unable to move. If the guess is too high we may not be able to construct a solar array large enough to move it.

Mass & Motors

Electric motors are rated by their torque output. Thus to choose a motor it is necessary to know how much torque is needed to move the bug. Torque, however, is mass dependant. Thus it is desirable to create an expression that gives motor torque as a function of mass. Once we have this expression we can chart the mass versus the motor torque required.

The force that is required to move the bug should be the sum of the forces to move each one of its parts. The bug will have two primary drive wheels and a small caster wheel to provide three points of support. The caster will be small and is for the moment assumed to be negligible. Thus the total force is represented by the following equation.

$$F_{total} = F_{bug} + 2(F_{wheel})$$

Equation 1

F_{total} in this case is the sum of the inertial force needed to move the bug plus the force required to roll both wheels. The inertial force to move the bug is represented by Equation 2 and the force to move a single wheel is given in Equation 3.

$$F = m \times a$$

Equation 2

$$F = \left(\frac{r}{R}\right) \times m \times g \times \mu_k$$

Equation 3

In Equation 3 R is the radius of the entire wheel and r is the radius of the motor shaft. Substituting these equations into the total force equation yields the following equation.

$$F_{total} = m_{bug} \times a + 2 \times \left(\frac{r}{R} \right) \times m_{wheel} \times g \times \mu_k$$

Equation 4

Applying the definition of torque found in Equation 5 to Equation 4 results in Equation 6.

$$\tau = F \times R$$

Equation 5

$$\tau_{wheel} = m_{bug} \times a \times R + 2 \times r \times m_{wheel} \times g \times \mu_k$$

Equation 6

This torque is accurate in reference to the radius of the wheel. This however, is not applicable to the torque needed at the motor shaft. To get motor shaft torque the torque applied on the wheel must be multiplied by the mechanical disadvantage between the wheel and the shaft as shown in Equation 7.

$$\tau_{motor} = \tau_{wheel} \times \left(\frac{R}{r} \right)$$

Equation 7

The final expression gives motor torque in terms of all of its sub components as seen in Equation 8.

$$\tau_{motor} = \left(\frac{m_{bug} \times a \times R^2}{r} \right) + 2 \times R \times m_{wheel} \times g \times \mu_k$$

Equation 8

This equation lets us define the wheel and shaft radius as well as the mass of each component. The constant μ_k is a constant for kinetic friction between the wheel and the floor. Entering in this function into excel with defined variables yields the following table.

Table 1

Variables	Value	Units
Wheel Radius	0.02	Meter
Wheel Mass	0.01	Kg
Acceleration	0.154	m/s
Shaft Radius	0.002	Meter
μ_k	0.4	

Needed Torque

Mass	Torque Wheel		Torque Shaft	
[kg]	[mNm]	[oz in]	[mNm]	[oz in]
0.1	0.46496	0.07	4.6496	0.6584
0.2	0.77296	0.11	7.7296	1.0946
0.3	1.08096	0.15	10.8096	1.5308
0.4	1.38896	0.20	13.8896	1.9669
0.5	1.69696	0.24	16.9696	2.4031
0.6	2.00496	0.28	20.0496	2.8393
0.7	2.31296	0.33	23.1296	3.2754
0.8	2.62096	0.37	26.2096	3.7116
0.9	2.92896	0.41	29.2896	4.1478
1	3.23696	0.46	32.3696	4.5839

Table 1 shows a range of masses from 100g to 1kg in increments of 100g. The assumptions for wheel radius, and shaft radius are based on the initial ideas of scale. The coefficient of friction is hard to define. There is no defined coefficient of friction for every specific material nor can it be derived outside of experimentation. I chose the coefficient of friction between two pieces of wood. The acceleration equates to roughly 6 in/s^2. This was determined to be an acceptable acceleration for the bug and is purely a matter of preference.

From this data it was decided to shoot for a mass allowance of 0.5kg. From Table 1 it is shown that this would require approximately 17 mNm of torque output from the motor. 17 mNm will be the approximate minimum motor torque. Electric motors are not entirely efficient due to loss to electromagnetic fields on the motor coils. To get 17 mNm from a motor directly is not difficult but requires more current than alternative methods. Current goes up proportionately with torque output thus to minimize current the motor must output as little torque as possible. To reduce the amount of motor torque required to move the bug a gearbox will be added to each motor. This gearbox will trade motor rpm for increased torque. This is a free trade off because most small electric motors run at several thousand rpm that would be impractical for this application. Gearing down the motor will not only provide more torque but slow the bug down as well.

$$\tau_{gs} = \tau_{motor} \times [GearRatio] \times [Efficiency]$$

Equation 9

Equation 9 represents the torque on the output of the gear shaft. To find out how much torque will be applied to the wheel the standard motor torque must be multiplied by the gear ratio. If the ratio is 100:1 then the motor torque is multiplied by 100. The result is then multiplied by the gear train efficiency to take into account energy loss in the gears themselves. This provides an accurate picture of how much torque is produced by the system. As stated before the goal torque is 17 mNm.

Gearboxes exist for many types of electric motors. For this application standard motors as well as stepper motors were considered for the primary drive system. The advantage with the stepper motors is that they have intrinsic control value. The exact number of rotations and speed can all be easily controlled. This is a preferable method to standard electric motors. However, stepper motors are far heavier than standard motors and require far greater operational current. It was discovered that the addition of an encoder could transform an otherwise open loop system to a closed loop feedback situation. Using an encoder on the standard electric motor will not provide extremely precise controls but will give information to at least one rpm. This is more than adequate for the application and thus was determined to be the best method.

The chosen motor system involves a standard electric motor with an associated gearbox and encoder. The best scenario would be to have the three components come as a set and be preassembled as to provide maximum efficiency between components and lessen construction time. A preliminary vendor has been identified and a motor assembly has been matched to the specifications. Maxon Motors manufactures the RE-16 electric motor. This motor has an associated gearbox and encoder that was made to work with the motor. The associated data sheets can be found in Appendix A.

This motor was chosen from the entire Maxon line because it was one of the few motors that had both the gear set and encoder system. This motor also had the highest torque/amp ratio of the other motors in consideration at 4.13 mNm/Amp. Using this ratio and the geared torque equation in Equation 10 the amount of current needed to run the setup can be calculated.

$$I_{operatoinal} = \left(\frac{\tau_{desired}}{GearRatio}\right) \times \left(Amp/Torque\right) \times \left[1 + (1 - Efficiency)\right]$$

Equation 10

For the RE-16 the amp/torque ratio is 1amp to 4.13 mNm at 3 volts. The motor assembly also has a gearing ratio of 84:1 that has an efficiency of 73%. Once again the desired output torque is 17 mNm. Inserting these values into Equation 10 yields Equation 11.

$$I_{operational} = \left\lfloor \frac{17mNm}{84} \right\rfloor \times \left\lfloor \frac{1A}{4.13mNm} \right\rfloor \times \left[1 + (1 - .73)\right]$$

Equation 11

This makes $I_{operational}$ 62.2 mA. For simplicity this is rounded to 65 mA. This means with a two motor setup the bug will draw 130mA when both motors are running. The motors are assumed to run at 3.3V in this scenario. This however does not have to be the rule. The RE-16, as with most electric motors, can be run at a variety of voltage levels that may change this current requirement. However, most low power 8051 microcontrollers operate at 3.3V. Thus the initial estimate is for a 3.3V power system.

Solar Cells and Power Systems

Solar cells are rated in terms of their efficiency, which is the percentage of the incident light power that is converted to electricity. Commercially-available cells range from below 10% efficient to 27% or more, but typical high-efficiency cells are around 20% efficient. This efficiency number is then applied to the amount of incident light power, called "insolation". This is commonly referred to in terms of an air-mass (AM) number. AM0 is equivalent to full sunlight at the edge of Earth's atmosphere, and is equal to 1367 W/m^2. AM1 is full sun with the sun directly overhead, equal to 925 W/m^2, and AM1.5 is full sun at 45 degrees from directly overhead, equal to 844 W/m^2. AM1.5 is typically used as the average value for incident sunlight over the course of the day, so a 20% efficient cell could be expected to produce 168.8 W/m^2 on average.

However, ambient lighting conditions are rarely as good as AM1.5. Heavy cloud cover can take insolation down to 20% of AM1.5, and insolation from indoor lighting can be as low as 2% to 0.2% of AM1.5. This would take our power production down to 33.76 W/m^2 for heavy cloud cover, and 0.3376 W/m^2 for minimal indoor lighting. Therefore, the design of the power supply is necessarily a trade-off between panel area and length of continuous operation under expected conditions.

To accommodate the variable nature of the solar cells, the array will be connected to a regulation circuit, which will supply a steady output voltage at variable current. Sharp spikes in current will be supplied by a capacitor connected in parallel with the regulated output. These spikes will primarily occur as a result of switching the motors on and off, so the intensity of the spikes will be predictable. The amplitude and duration of the spikes will determine the size of capacitor needed and the maximum switching rate of the motors. Longer-term current shortages could arise from insufficient insolation on the solar panels. The necessary current would be supplied by a backup battery, which would be charged during the periods when excess power is available from the solar panels.

Since power regulation is taken care of, we need only consider operating conditions and physical constraints in the array design. It must be possible to arrange the cells to give roughly a 3.3V output. These 3.3V series modules can then be connected in parallel to increase current output. For example, the SunPower Pegasus cells produce 0.565V and 452mA per cell at AM1.5. These would need to be arranged in modules of 6 to get 3.4V unregulated. If we assume the total current requirements of the bug to be 180mA to make the math easy, then we can determine the operating conditions from the array size. Two modules would produce 180mA under heavy cloud cover conditions described above; twenty modules would be needed to produce 180mA under strong indoor lighting. Each cell is 21.96 cm^2, so two modules would be 263 cm^2 and twenty would be 2635 cm^2. 263 cm^2 is just over 16cm x 16cm, but 2635cm^2 is 51cm x 51cm, which is probably too large.

As a practical matter, the array design will consist of constructing as large and efficient an array as acceptable, and calculating the minimal continuous operation point (and duty cycles for lower light levels) from that.

Schedule

Spring Semester		
Week 1	Solar Panel Testing	Motor Testing
Week 2	Power Regulation	Motor Control Hardware
Week 3	Power Regulation	Motor Control Hardware
Week 4	Power Regulation	Motor Hardware Testing
Week 5	Solar Panel and Battery Integration	Motor Power System Integration
Week 6	Optoelectronic Testing Microcontroller setup	Optoelectronic Testing Microcontroller setup
Week 7	Software Modules	Software Modules
Week 8	Software Modules	Software Modules
Week 9	Software Modules	Software Modules
Week 10	Software Testing	Software Testing
Week 11	Final Product Assembly	Final Product Assembly
Week 12	Product Testing	Product Testing
Week 13	Product Testing	Product Testing
Week 14	Finishing Touches	Finishing Touches

Equipment List

- N solar panels (number and vendor to be determined)
- N bypass diodes (for solar panel)
- Back-up battery (CR2032)
- Power-supply regulator
- Large power regulation capacitor
- Phytec phyCORE microMODUL 8051 LP board
- Two Maxon RE-16 motors with gear trains, encoders, and wheels
- Mounting platform
- 1 or more Collision sensors
- IR transmitter (diode)
- 4 IR receivers (IR-sensitive transistors)
- 2 light detectors
- Caster wheel
- 2 or more LEDs for status
- Buttons for user input (switch-type, return to open)
- Miscellaneous discrete components (resistors, diodes, etc.)

Patent Survey

A patent search was performed at http://www.uspto.gov/ for various keywords relating to the proposed robot: solar power, autonomous, robot, photovoltaic, etc. The following are the most relevant patents found.

- 6,338,013, 6,502,017, 6,600,981, 6,611,738, 6,650,975. Essentially identical to each other, these patents describe a lawn-care appliance with an optional photovoltaic charger system.

- 5,554,914, 5,610,488. These describe a very small scale (1 cm^3) robot for small physical tasks (medical examination and miniaturized assembly, for example), where a PV element is used for power and data transmission.

- 5,869,910. Describes a power system for self-contained mobile robots, but the primary focus of the patent is on wireless power transmission using RF; PV cells are in effect the competition here.

- 4,777,416. Describes a robot which can join with a charging station as necessary.

- 4,767,334. Describes a very complex and large (big enough to hold a small child) toy robot vehicle. The PV element is rather bizarre; the user shines a penlight on it in order to provide the power to illuminate a display panel. It's unclear why they would waste the energy required to turn electricity to light to electricity when a simple switch would suffice.

- 6,648,720. Describes a clever locomotion method for small robots, wherein outboard legs are supplemented by an inboard drive wheel.

- 6,459,955. Describes a home-cleaning robot with lawn-care mode, with optional solar charging.

- 6,160,371. Describes operator-controlled distributed control of autonomous robots (with, of course, optional solar cells for power).

- 5,897,156. Describes a process in which a rover can react to a hostile environment (eg, sandstorms, long nights).

Some of these patents use similar to our project none of them directly apply. There is little worry that an existing patent will be violated with this project.

Standards

- The microcontroller conforms to the industry-standard 8051 architecture.
- Air Mass (AM) numbers are a shorthand for the incident power and spectrum of light. AM0 is 1367 W/m^2 and 5800K black-body radiation spectrum; AM1.5 is 844 W/m^2 and 5600K black-body spectrum.
- Standard test conditions for solar cells are: 1000 W/m^2 insolation, equivalent spectrum of AM1.5 (equivalent to a 5600K black-body radiator), with unit temperature of $25^{O}C$. These conditions are assumed on datasheets if no additional information is given.

Bibliography

Buresch, Matthew, Photovoltaic Energy Systems. New York: 1982, McGraw-Hill Book Company.

Messenger, Roger and Ventre, Jerry, Photovoltaic Systems Engineering. Boca Raton: 2000, CRC Press.

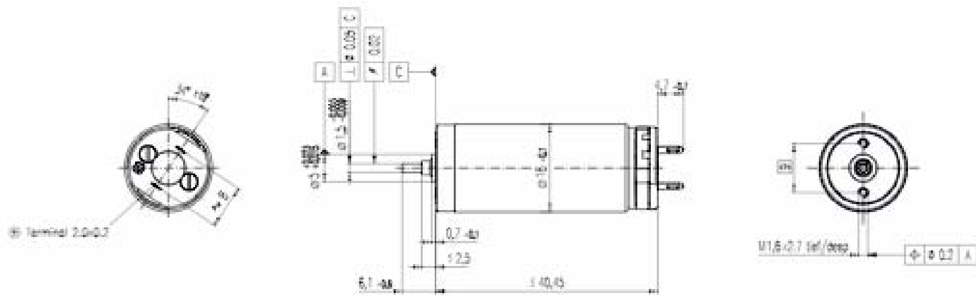
Morecock, Earle M., Direct-Current Circuits. New York: 1953, McGraw-Hill Book Company.

Rauschenbach, H. S., Solar Cell Array Design Handbook. New York: 1980, Van Nostrand Reinhold Company.

Reese, Ronald Lane, University Physics. Pacific Grove: 2000, Brooks/Cole Publishing Company.

Appendix A

RE 16 ∅16 mm, Precious Metal Brushes CLL, 3.2 Watt, CE approved



Stock program
Standard program
Special program (on request!)

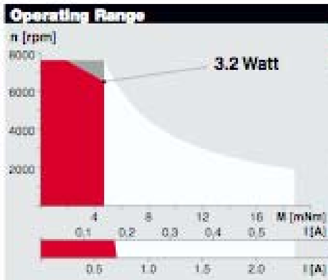
Order Number

Motor Data	Unit	118678	118679	118680	118681	118682	118683	118684	118685	118686	118687	118688	118689	118690	118691	118692
1 Assigned power rating	W	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2
2 Nominal voltage	Volt	1.8	2.4	3.0	3.2	4.5	4.8	7.2	9.0	12.0	12.0	15.0	18.0	24.0	30.0	48.0
3 No load speed	rpm	4990	6360	6890	6270	6740	5700	6890	6740	7130	5990	6010	5900	7250	6460	5500
4 Stall torque	mNm	15.5	16.9	17.3	15.2	17.4	15.2	18.1	17.4	18.6	14.9	14.9	14.5	17.6	15.0	11.9
5 Speed / torque gradient	rpm / mNm	323	379	400	415	391	378	383	389	386	404	408	410	414	432	465
6 No load current	mA	24	25	23	19	15	11	10	7	6	5	4	3	3	2	1
7 Starting current	mA	4530	4710	4190	3130	2740	1900	1820	1370	1160	794	628	500	561	341	144
8 Terminal resistance	Ohm	0.397	0.510	0.715	1.02	1.64	2.53	3.95	6.56	10.3	15.3	23.9	36.0	42.8	88.0	333
9 Max. permissible speed	rpm	7600	7600	7600	7600	7600	7600	7600	7600	7600	7600	7600	7600	7600	7600	7600
10 Max. continuous current	mA	720	720	720	720	720	708	566	439	350	297	230	187	172	120	61.6
11 Max. continuous torque	mNm	2.47	2.58	2.98	3.49	4.56	5.66	5.62	5.57	5.59	5.47	5.45	5.42	5.40	5.28	5.09
12 Max. power output at nominal voltage	mW	2020	2800	3120	2490	3060	2260	3260	3060	3460	2330	2330	2230	3340	2530	1710
13 Max. efficiency	%	86	86	86	85	86	86	86	86	86	85	85	85	86	85	84
14 Torque constant	mNm / A	3.43	3.58	4.13	4.84	6.34	7.99	9.92	12.7	16.0	19.0	23.7	28.9	31.4	44.1	82.7
15 Speed constant	rpm / V	2790	2660	2310	1970	1510	1190	962	753	597	502	403	330	304	217	115
16 Mechanical time constant	ms	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5
17 Rotor inertia	gcm²	1.65	1.38	1.27	1.20	1.25	1.27	1.25	1.22	1.23	1.17	1.17	1.15	1.15	1.10	1.02
18 Terminal inductance	mH	0.02	0.02	0.03	0.04	0.07	0.11	0.17	0.28	0.45	0.64	0.99	1.48	1.75	3.44	12.10
19 Thermal resistance housing-ambient	K / W	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30
20 Thermal resistance rotor-housing	K / W	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5	8.5
21 Thermal time constant winding	s	14	11	10	10	10	10	10	10	10	10	9	9	9	9	8

Specifications

- Axial play 0.05 - 0.15 mm
- Max. **sleeve bearing** loads
  axial (dynamic) 0.8 N
  radial (5 mm from flange) 1.5 N
  Force for press fits (static) 15 N
- Radial play **sleeve bearing** 0.014 mm
- Ambient temperature range -20 ... +65°C
- Max. rotor temperature +85°C
- Number of commutator segments 7
- Weight of motor 38 g
- Values listed in the table are nominal.
  For applicable tolerances see page 43.
  For additional details please use the maxon selection program on the enclosed CD-Rom.
- CLL = Capacitor Long Life

Operating Range



Comments Details on page 49

Recommended operating range

Continuous operation
In observation of above listed thermal resistances (lines 19 and 20) the maximum permissible rotor temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

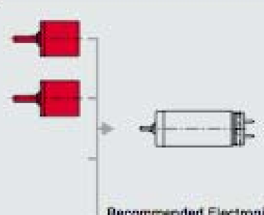
Short term operation
The motor may be briefly overloaded (recurring).

118690 Motor with high resistance winding
118683 Motor with low resistance winding

maxon Modular System Overview on page 17 - 21

Planetary Gearhead
∅16 mm
0.1 - 0.3 Nm
Details page 185

Planetary Gearhead
∅16 mm
0.06 - 0.18 Nm
Details page 186

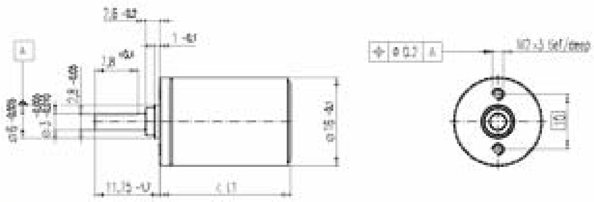


Recommended Electronics:
LSC 30/2 page 231
Notes 17

April 2003 edition / subject to change

Planetary Gearhead GP 16 A ∅16 mm, 0.1 - 0.3 Nm

Metal Version



M 1:1

Technical Data

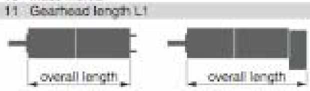
Planetary Gearhead	straight teeth
Output shaft	stainless steel, hardened
Bearing at output	sleeve bearings*
Radial play, 6 mm from flange	max. 0.06 mm
Axial play	0.02 - 0.10 mm
Max. permissible axial load	8 N
Max. permissible force for press fits	100 N
Recommended input speed	< 6000 rpm
Recommended temperature range	-15 ... +65°C

Number of stages	1	2	3	4	5
Max. perm. radial load, 6 mm from flange	8 N	12 N	16 N	20 N	20 N

*Options: Ball bearing

Stock program
Standard program
Special program (on request!)

Order Number		110321	110322	110323	118186	110324	134782	110325	134785
Gearhead Data									
1 Reduction		4.4 : 1	19 : 1	84 : 1	157 : 1	370 : 1	690 : 1	1621 : 1	3027 : 1
2 Reduction absolute		57/13	3249/169	185193/2197	19683/125	10556001/28561	1121931/1625	60169057/37129	63950067/21125
Order Number		118184	134777	134778		134780	118187	134783	134786
1 Reduction		5.4 : 1	24 : 1	104 : 1		455 : 1	850 :1	1996 : 1	3728 : 1
2 Reduction absolute		27/5	1539/64	87723/845		5000211/10985	531441/625	28672327/14365	30292137/8125
Order Number			118185	134779		134781		134784	118188
1 Reduction			29 : 1	128 : 1		561 : 1		2458 : 1	4592 : 1
2 Reduction absolute			729/25	41553/325		2368521/4225		135005697/54925	14348907/3125
3 Number of stages		1	2	3	3	4	4	5	5
4 Max. continuous torque at gear output	Nm	0.10	0.15	0.20	0.20	0.25	0.25	0.30	0.30
5 Intermittently permissible torque at gear output	Nm	0.150	0.225	0.300	0.300	0.375	0.375	0.450	0.450
6 Sense of rotation, drive to output		=	=	=	=	=	=	=	=
7 Max. efficiency	%	90	81	73	73	65	65	59	59
8 Weight	g	20	23	27	27	31	31	35	35
9 Average backlash no load	°	0.7	0.8	1.0	1.0	1.2	1.2	1.5	1.5
10 Mass inertia	gcm²	0.08	0.05	0.05	0.05	0.05	0.05	0.05	0.05
11 Gearhead length L1	mm	15.5	19.1	22.7	22.7	26.3	26.3	29.9	29.9

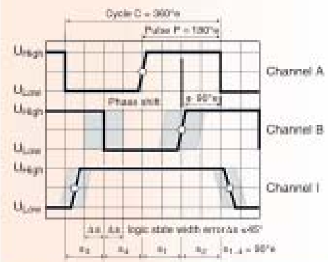


Combination:

+ Motor	Page	+ Tacho / Encoder / Brake	Page	Overall length [mm] = Motor length + gearhead length + (tacho / encoder / brake) + assembly parts							
RE 15, 1.6 W	71			37.9	41.5	45.1	45.1	48.7	48.7	52.3	52.3
RE 16, 3.2 W	72 / 73			56.0	59.6	63.2	63.2	66.8	66.8	70.4	70.4
RE 16, 3.2 W	73	MR Encoder	210/211	61.0	64.6	68.2	68.2	71.8	71.8	75.4	75.4
RE 16, 3.2 W	73	Digital Magnetic Encoder 13	223	62.1	65.7	69.3	69.3	72.9	72.9	76.5	76.5
RE 16, 4.5 W	74 / 75			59.0	62.6	66.2	66.2	69.8	69.8	73.4	73.4
RE 16, 4.5 W	75	MR Encoder	210/211	64.0	67.6	71.2	71.2	74.8	74.8	78.4	78.4
RE 16, 4.5 W	75	Digital Magnetic Encoder 13	223	65.1	68.7	72.3	72.3	75.9	75.9	79.5	79.5
A-max 16	101-104			41.0	44.6	48.2	48.2	51.8	51.8	55.4	55.4
A-max 16	102/104	MR Encoder	210/211	46.0	49.6	53.2	53.2	56.8	56.8	60.4	60.4
A-max 16	101/104	Digital Magnetic Encoder 13	223	49.1	52.7	56.3	56.3	59.9	59.9	63.5	63.5
RE-max 17	131-134			41.0	44.6	48.2	48.2	51.8	51.8	55.4	55.4
RE-max 17	132/134	MR Encoder	210/211	46.0	49.6	53.2	53.2	56.8	56.8	60.4	60.4
EC 16, 15 W	150			55.8	59.4	63.0	63.0	66.6	66.6	70.2	70.2

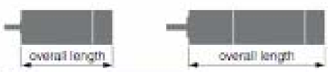
Digital MR Encoder, Type M

maxon tacho



Stock program
Standard program
Special program (on request!)

Order Number	201935	201936
Type		
Counts per turn	32	32
Number of channels	2	3
Max. operating frequency (kHz)	8	8



Combination

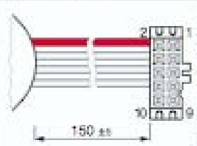
+ Motor	Page	+ Gearhead	Page	+ Brake	Page	Overall length [mm] / ● see: + Gearhead	
RE 16, 3.2 W	73					45.4	45.4
RE 16, 3.2 W	73	GP 16, 0.1 - 0.3 Nm	185			●	●
RE 16, 3.2 W	73	GP 16, 0.06 - 0.18 Nm	186			●	●
RE 16, 4.5 W	75					48.4	48.4
RE 16, 4.5 W	75	GP 16, 0.1 - 0.3 Nm	185			●	●
RE 16, 4.5 W	75	GP 16, 0.06 - 0.18 Nm	186			●	●
A-max 16	102/104					30.4	30.4
A-max 16	102/104	GS 16, 0.015 Nm	181			●	●
A-max 16	102/104	GS 16, 0.01 - 0.1 Nm	182-184			●	●
A-max 16	102/104	GP 16, 0.1 - 0.3 Nm	185			●	●
A-max 16	102/104	GP 16, 0.06 - 0.18 Nm	186			●	●
A-max 19	106					34.0	34.0
A-max 19	106	GP 19, 0.1 - 0.3 Nm	187			●	●
A-max 19	106	GP 22, 0.5 - 2.0 Nm	190/191			●	●
A-max 19	106	GS 24, 0.1 Nm	192			●	●
A-max 19	108					35.8	35.8
A-max 19	108	GP 19, 0.1 - 0.3 Nm	187			●	●
A-max 19	108	GP 22, 0.5 - 2.0 Nm	190/191			●	●
A-max 19	108	GS 24, 0.1 Nm	192			●	●
A-max 22	110/112					36.9	36.9
A-max 22	110/112	GP 22, 0.1 - 0.6 Nm	188/189			●	●
A-max 22	110/112	GP 22, 0.5 - 2.0 Nm	190/191			●	●
A-max 22	110/112	GS 24, 0.1 Nm	192			●	●
RE-max 17	132/134					30.4	30.4
RE-max 17	132/134	GP 16, 0.1 - 0.3 Nm	185			●	●
RE-max 17	132/134	GP 16, 0.06 - 0.18 Nm	186			●	●
RE-max 21	136					34.0	34.0
RE-max 21	136	GP 22, 0.5 - 2.0 Nm	190/191			●	●
RE-max 21	136	GS 38, 0.1 - 0.6 Nm	200			●	●
RE-max 21	138					35.8	35.8
RE-max 21	138	GP 22, 0.5 - 2.0 Nm	190/191			●	●
RE-max 21	138	GS 38, 0.1 - 0.6 Nm	200			●	●
RE-max 24	140/142					36.9	36.9
RE-max 24	140/142	GP 22, 0.5 - 2.0 Nm	191			●	●
RE-max 24	140/142	GS 38, 0.1 - 0.6 Nm	200			●	●

Technical Data

Supply voltage V_{CC}	2.5 - 5.5 V
Output signal at V_{CC} = 5 VDC	TTL compatible
Index pulse width (nominal)	> 90°e
Operating temperature range	-25 ... +85°C
Moment of inertia of code wheel	≤ 0.09 gcm²
Output current per channel	max. 5 mA

Attention: The index signal I is not synchronised with channel A or B.
The length of the index signal can last more than one cycle.

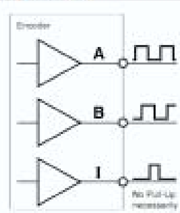
Pin Allocation



1 Motor +
2 V_{CC}
3 Channel A
4 Channel B
5 Gnd
6 Motor –
7* Channel I (Index)

DIN Connector 41651
Flat cables AWG 28
*version with 3 channels

Connection Example



April 2003 edition / subject to change