

1 Introduction

In this app note we will outline how to make a smart pillow that is sensitive to snoring, using a GreenPAK™ SLG46620.

The smart pillow relies upon vibration to indicate to the sleeper when he snores while sleeping. It works automatically when a person puts his head on the pillow.

Snoring is an unfortunate condition because it affects not only on the snoring person but also people who are sleeping around him. Snoring has been voted as the greatest medical reason behind divorce in the USA. Additionally, sleep apnea can cause a wide array of health problems that can be mitigated by ensuring a sleeper does not choose a position that leads to snoring.

In this app note, we will build a system which can detect & analyze sounds. When it analyzes a snoring sound, it will turn on a vibration motor so that the sleeper wakes up.

When the sleeping person lifts his head off the pillow, the vibration motor will stop. When a sleeper changes their sleeping position they are more likely to settle into a different position that will prevent snoring.

Pillow tasks:

- The pillow has a touch sensor so that the system is enabled automatically when the person places his/her head on pillow, and is idle when he raises his head up.
- When the system detects a snoring sound or any other cacophonous sound, a vibrator is turned on to wake the sleeper up.
- Features 2 user settable vibration modes: continuous or pulsed.

The system is useful for people who suffer from snoring. For safety, people who suffer from very deep sleep can also use the system because it can detect doorbells, ringing phones or crying babies.

We implemented this project with a SLG46620 GreenPAK IC, a sound sensor, a vibration motor, force-sensing resistor and some passive components.

The total number of components for this design is quite minimal, despite not utilizing a microcontroller. Since GreenPAK IC's are low-cost and have low-power consumption, they are an ideal component for this solution. Their small size also would allow them to be easily integrated inside the pillow without manufacturing concerns.

Most projects that depend on sound detection have a "false trigger rate", which is necessary due to the possibility of error amongst a variety of sensors. The sensors associated with this project merely detect a level of sound; they do not detect the type of sound or the nature of its origin. Consequently, a false trigger can be caused by an act such as clapping, knocking or other noise unrelated to snoring that may be detected by the sensor.

In this project the system will ignore the short sounds which cause false trigger rate, so we will build a digital filter that can detect a sound segment like the sound of snoring.

Look at the following graphic curve in figure 1 which represents the sound of snoring.

We can see that it consists of two sections which are repeated and time correlated.

The first section detects snoring; it is a sequence of short pulses that lasts for 0.5 to 4 seconds, followed by a silence period which lasts for 0.4 to 4 seconds and may contain background noise.

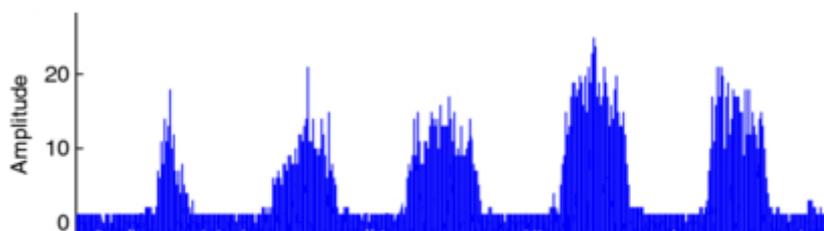


Figure 1. Spectrum of snoring sound

Therefore, to filter out other noises the system must detect a snoring segment, which lasts for more than 0.5 sec, and ignore any briefer sound segment.

To make the system more stable, a counter should be implemented which counts the snoring segments to launch the alarm after the detection of two sequential snoring segments.

In this case, even if a sound lasts for more than 0.5 sec, the system will filter it unless it is repeated within a specific time frame. In this way, we can filter the sound that can be caused by a movement, cough or even short noise signals.

2 Implementation Plan:

The design of this project consists of two sections; the first section is responsible for detecting sound and analyzes it to detect the sound of snoring to alert the sleeper.

The second section is a touch sensor; it is responsible for automatically enabling the system when a person puts his head on the pillow, and to disable the system when the sleeping person lifts his head off the pillow.

Whenever a person puts their head on the pillow, the touch sensor sends an activation signal from Matrix2 to Matrix1 through P10 to activate the circuit and to start taking samples from the sound sensor.

The system takes a sample from the sound sensor every 30ms within a 5ms time frame. In this way, energy consumption will be saved & short sound pulses will be filtered.

If we detect 15 sequential sound samples (no silence lasts for more than 400ms between any of the samples), it is concluded that the sound is persistent. In this case, the sound segment will be considered a snoring segment. When this action repeats after a silence, which lasts more than 400ms and less than 6s, the captured sound will be considered snoring and the sleeper will be alerted by vibration.

You can delay warning for more than 2 snoring segments to raise accuracy from pipedelay0 configuration in the design, but this may increase response time. 6sec frame would need to be increased as well.

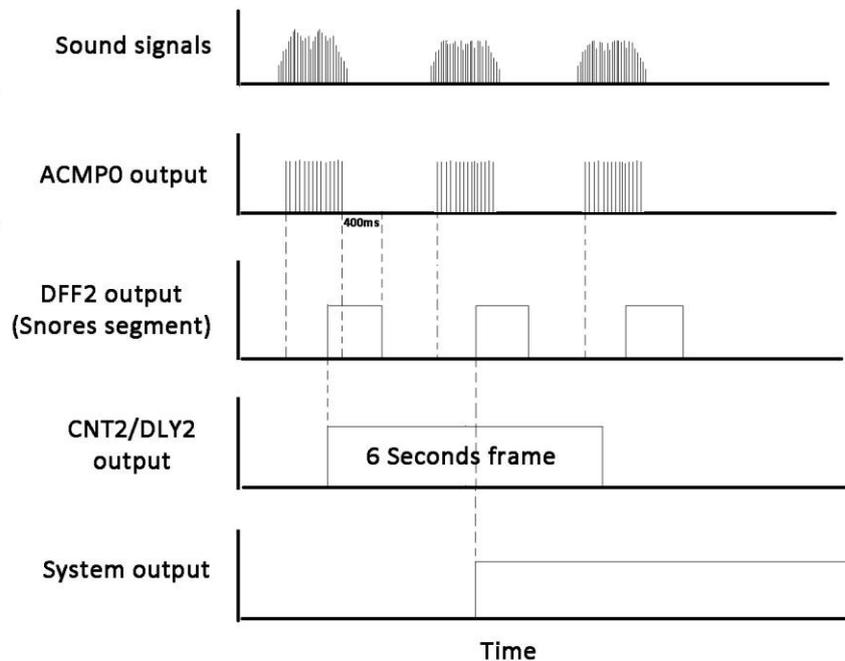


Figure 2. Sampling and analysis process

LUT3 is used to reset the Pipedelay, so its output will be Low if the sleeping person lifts their head off the pillow. In this case, the time gate of CNT2 is finished before detecting two sequential snoring segments.

Pin3 is configured as an input & is connected to a "Vibration mode button". The signal coming from pin3 passes through DFF4 and DFF5 configures the vibration pattern to one of two patterns: mode1 and mode2. In case of mode1: when snoring is detected, a continuous signal is sent to the vibration motor, which means that the motor runs continuously.

In the case of mode2: when snoring is detected, the vibration motor is pulsed with the timing of CNT6 output. So when the output of DFF5 is high, mode1 will be activated. When it is low(mode 2), the output of DFF4 is high, and the output of CNT6 will appear on pin9 through LUT1.

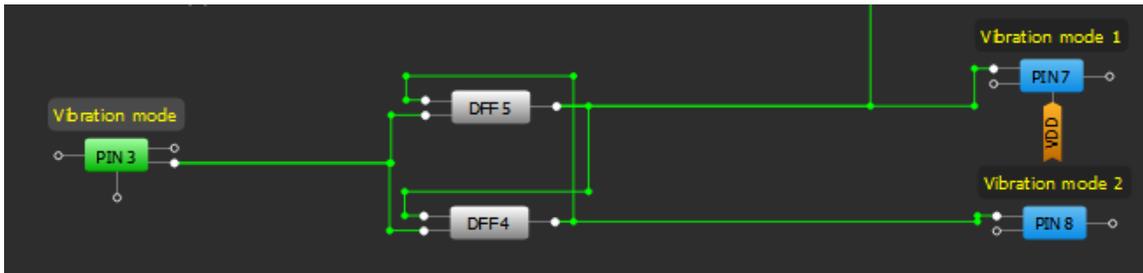


Figure 4. Vibration mode saving blocks

Sensitivity to the sound sensor is controlled by a potentiometer which is set in the module. The sensor should be initialized manually for the first time to get the required sensitivity.

PIN10 is connected to the output of ACMP0, which is externally connected to an LED. When the sound sensor is calibrated, the output of pin10 should be quite low, which means that there is no flicker on the external LED which is connected to pin10. In this way, we can guarantee that the voltage which is generated by the sound sensor in silence doesn't exceed the 300mv ACMP0 threshold.

If you need another alarm in addition to vibration, you can connect a buzzer to pin9 so that a sound alarm will be activated as well.

4 Second Section: Touch Sensor

The touch sensor that we built uses Force-sensing resistor (FSR). Force-sensing resistors consist of a conductive polymer which changes resistance in a predictable manner following application of force to its surface. The sensing film consists of both electrically conducting and non-conducting particles suspended in a matrix. Applying a force to the surface of the sensing film causes particles to touch the conducting electrodes, changing the resistance of the film. FSR comes with different sizes and shapes (circle & square).



Figure 5. Force-sensing resistor (FSR)

The resistance exceeded 1 MΩ with no applied pressure and ranged from around 100 kΩ to a few hundred Ohms as a pressure varied from light to heavy. In our project, FSR will be used as head touch sensor and it is located inside the pillow. The average human head weight is between 4.5 and 5kg. When the user puts his head on the pillow, a force is applied on the FSR and its resistance changes. GreenPAK detects this change and the system is enabled.

The way to connect a resistive sensor is to connect one end to Power and the other to a pull-down resistor to ground. Then the point between the fixed pull down resistor and the variable FSR resistor is connected to the analog input of a GreenPAK (Pin12) as shown in figure 7. The signal will be brought from the pin to the input of ACMP1. The other input of ACMP1 is connected to a 1200mv reference setting. Comparison result is stored in DFF6. When a head touch is detected, the output of DFF2 turns HI to activate CNT2/Dly2, which is configured to work as "falling edge delay" with a delay equal to 1.5 sec. In this case, if the sleeper moves or turns from side to side and the FSR is interrupted less than 1.5 sec, the system is still activated and no reset occurs. CNT7 and CNT8 are used to enable FSR and ACMP1 for 50 mS every 1sec in order to reduce power consumption.

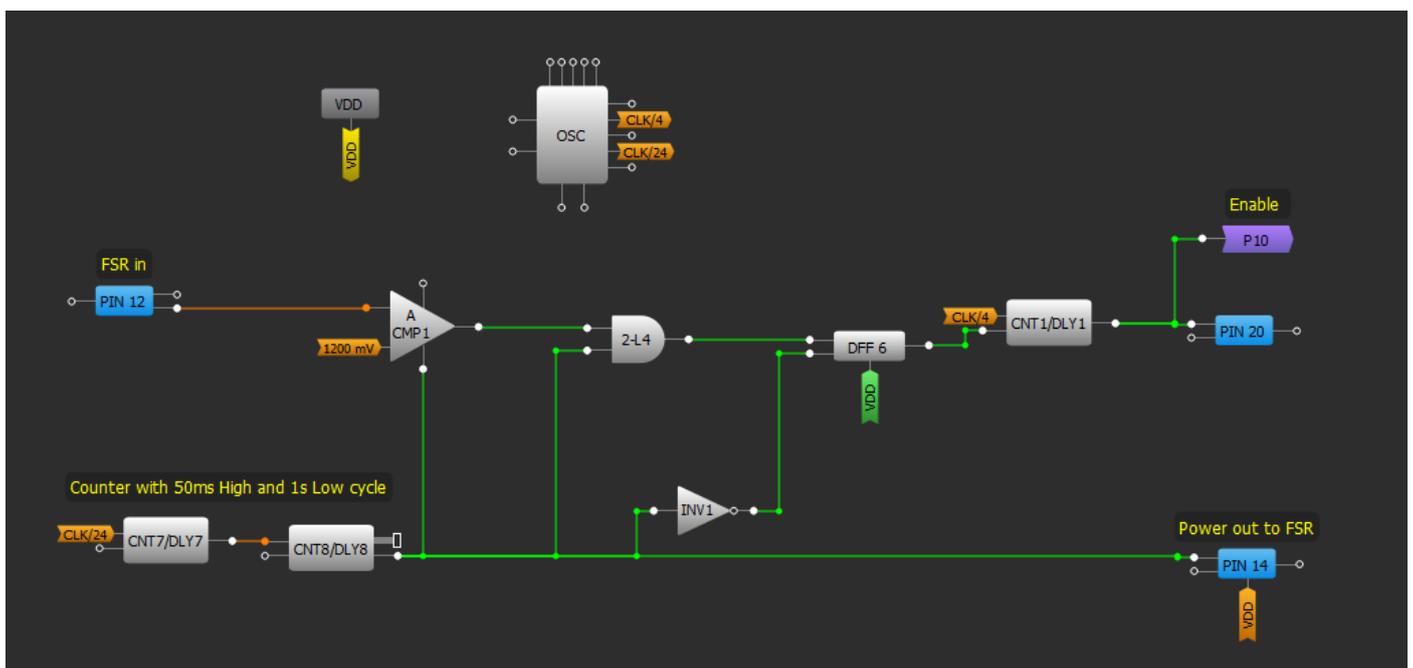


Figure 6. Touch sensor design

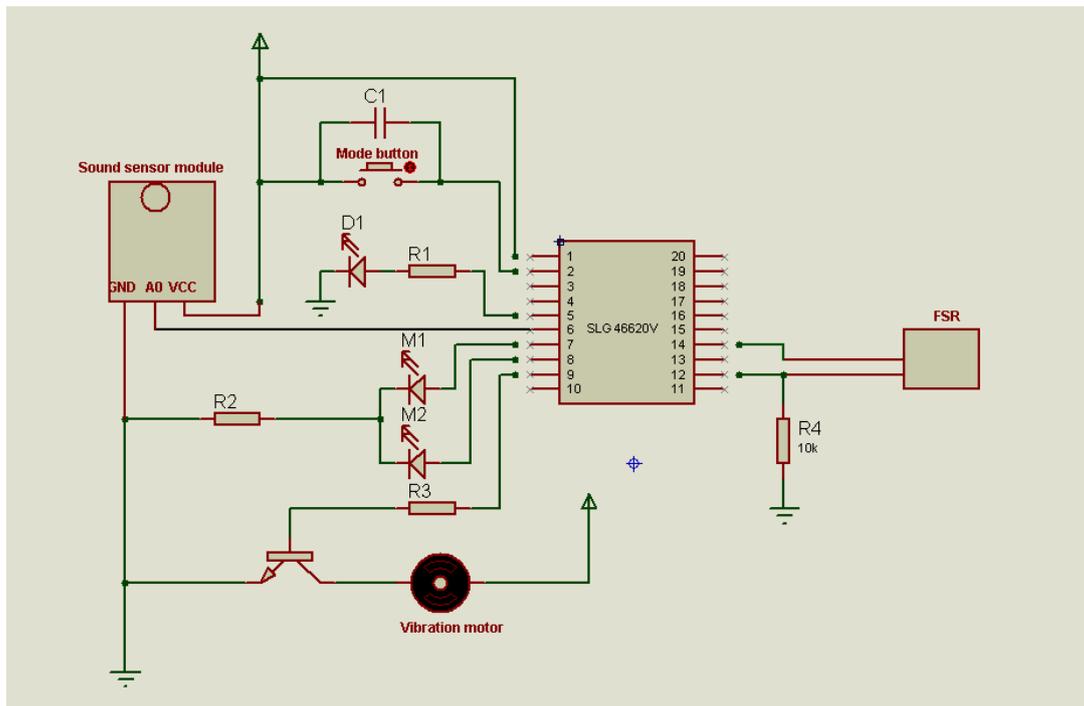


Figure 7. Top level circuit

5 Conclusion

In this project we made a smart pillow which is used for snoring detection to alert the sleeping person by vibration.

We also made touch sensor using FSR to activate the system automatically when using the pillow. A further enhancement option could be to design in parallel FSR's to accommodate larger size pillows. We also made digital filters to minimize the occurrence of false alarms. The GreenPAK IC was successfully configured to make digital sampling and filtration. It allowed us to easily integrate several discrete components from the touch sensor to the sound filters and detectors inside a very small physical area.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.